SafeNet Authentication Client 10.8 R2 (GA)

MAC ADMINISTRATOR GUIDE

Document Information

Document Information

Product Version	10.8 R2 (GA)
Document Number	007-013726-004
Release Date	May 2023

Revision History

Revision	Date	Reason
C	May 2023	Updated for 10.8 R2 (GA) release

Thales hereby disclaims all warranties and conditions with regard to the information contained herein, including all implied warranties of merchantability, fitness for a particular purpose, title and non-infringement. In no event shall Thales be liable, whether in contract, tort or otherwise, for any indirect, special or consequential damages or any damages whatsoever including but not limited to damages resulting from loss of use, data, profits, revenues, or customers, arising out of or in connection with the use or performance of information contained in this document.

Thales does not and shall not warrant that this product will be resistant to all possible attacks and shall not incur, and disclaims, any liability in this respect. Even if each product is compliant with current security standards in force on the date of their design, security mechanisms' resistance necessarily evolves according to the state of the art in security and notably under the emergence of new attacks. Under no circumstances, shall Thales be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Thales products. Thales disclaims any liability with respect to security for direct, indirect, incidental or consequential damages that result from any use of its products. It is further stressed that independent testing and verification by the person using the product is particularly encouraged, especially in any application in which defective, incorrect or insecure functioning could result in damage to persons or property, denial of service, or loss of privacy.

CONTENTS

PREFACE: About this Document

This document describes the operational and administrative tasks you can perform to maintain the functionality and efficiency of your SafeNet Authentication Client (SAC).

This section also identifies the audience, explains how to best use the written material, and discusses the documentation conventions used. They are:

> "Audience" below
> "Related Documents" on page 1
> "Document Conventions" below
> "Support Contacts" on page 8

For information regarding the document status and revision history, refer to "Document Information" on page 1.

Audience

This document is intended for personnel responsible for maintaining your organization's security infrastructure.

All products manufactured and distributed by Thales Group are designed to be installed, operated, and maintained by personnel who have the knowledge, training, and qualifications required to safely perform the tasks assigned to them. The information, processes, and procedures contained in this document are intended for use by trained and qualified personnel only.

It is assumed that the users of this document are proficient with security concepts.

Document Conventions

This section describes the conventions used in this document.

Command Syntax and Typeface Conventions

This document uses the following conventions for command syntax descriptions, and to highlight elements of the user interface.

Format	Convention
bold	The bold attribute is used to indicate the following: > Command-line commands and options that you enter verbatim (Type **dir /p**.) > Button names (Click **Save As**.) > Check box and radio button names (Select the **Print Duplex** check box.) > Dialog box titles (On the **Protect Document** dialog box, click **Yes**.) > Field names (**User Name**: Enter the name of the user.) > Menu names (On the **File** menu, click **Save**.) (Click **Menu** > **Go To** > **Folders**.) > User input (In the **Date** box, type **April 1**.)
italics	In type, the italic attribute is used for emphasis or to indicate a related document. (See the *Installation Guide* for more information.)
<variable>	In command descriptions, angle brackets represent variables. You must substitute a value for command line arguments that are enclosed in angle brackets.
[**optional**] [<optional>]	Represent optional **keywords** or <variables> in a command line description. Optionally enter the keyword or <variable> that is enclosed in square brackets, if it is necessary or desirable to complete the task.
{**a**|**b**|**c**} {<a>|<b>|<c>}	Represent required alternate **keywords** or <variables> in a command line description. You must choose one command line argument enclosed within the braces. Choices are separated by vertical (OR) bars.
[**a**|**b**|**c**] [<a>|<b>|<c>]	Represent optional alternate keywords or variables in a command line description. Choose one command line argument enclosed within the braces, if desired. Choices are separated by vertical (OR) bars.

Notifications and Alerts

Notifications and alerts are used to highlight important information or alert you to the potential for data loss or personal injury.

Tips

Tips are used to highlight information that helps to complete a task more efficiently.

> **TIP** This is some information that will allow you to complete your task more efficiently.

Notes

Notes are used to highlight important or helpful information.

> **NOTE** Take note. Contains important or helpful information.

Cautions

Cautions are used to alert you to important information that may help prevent unexpected results or data loss.

> **CAUTION!** Exercise caution. Contains important information that may help prevent unexpected results or data loss.

Warnings

Warnings are used to alert you to the potential for catastrophic data loss or personal injury.

> ****WARNING** Be extremely careful and obey all safety and security measures. In this situation you might do something that could result in catastrophic data loss or personal injury.**

Support Contacts

If you encounter a problem while installing, registering, or operating this product, please refer to the documentation before contacting support. If you cannot resolve the issue, contact your supplier or Thales Customer Support.

Thales Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between Thales and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Customer Support Portal

The Customer Support Portal, at https://supportportal.thalesgroup.com, is where you can find solutions for most common problems. The Customer Support Portal is a comprehensive, fully searchable database of support resources, including software and firmware downloads, release notes listing known problems and workarounds, a knowledge base, FAQs, product documentation, technical notes, and more. You can also use the portal to create and manage support cases.

> **NOTE** You require an account to access the Customer Support Portal. To create a new account, go to the portal and click on the **REGISTER** link.

Telephone

The support portal also lists telephone numbers for voice contact (Contact Us).

Email Support

You can also contact technical support by email at technical.support.DIS@thalesgroup.com.

CHAPTER 1:   Introduction

SafeNet Authentication Client (SAC) is a middleware client that manages Thales's extensive SafeNet portfolio of certificate-based authenticators, including eToken, IDPrime smart cards, USB, and software based devices.

With full backward compatibility and incorporating features from previous middleware versions, SAC ensures complete support for all currently deployed eToken as well as IDPrime smart cards.

> **NOTE** The term *Token* is used throughout the document and is applicable to both Smart Cards and USB Tokens.

Overview

SAC is a Public Key Infrastructure (PKI) middleware that provides a secure method for exchanging information based on public key cryptography, enabling trusted third-party verification of user identities. It utilizes a system of digital certificates, Certificate Authorities, and other registration authorities that verify and authenticate the validity of each party involved in an internet transaction.

SAC enables the implementation of strong two-factor authentication using standard certificates as well as encryption and digital signing of data. Generic integration with Keychain and PKCS#11 security interfaces enables out-of-the-box interoperability with a variety of security applications offering secure web access, PC and data security, and secure email. PKI keys and certificates are created, stored, and used securely from within hardware or software.

The SAC Tools application and the SAC tray icon application are installed with SAC, providing easy-to-use configuration tools for users and administrators.

> **NOTE** Both these applications (SAC Tools and SAC Monitor) work in *Light Mode* only.

For SAC system requirement details and compatibility information, refer to *SafeNet Authentication Client Release Notes*.

Password Quality Information

SAC supports password quality settings for Administrator Passwords (also known as Security Officer (SO) passwords) and Initialization keys that are implemented by SAC software. The setting is the same for all devices and cannot be modified. Though, it can be switched off for backward compatibility.

Additionally, IDPrime supports the insertion of the Administrator Key directly (without derivation), in which case the password policy is not validated. The Administrator Key derivation method is proprietary and may vary depending on the device.

The Administrator Password quality and Initialization Key quality must include three out of the following four rules:

1. English uppercase letters (ASCII 0x41...0x5A)

2. English lowercase letters (ASCII 0x61...0x7A)

3. Numeric (ASCII 0x30...0x39)

4. Special characters (ASCII 0x20...0x2F + 0x3A...0x40 + 0x5B...0x60 + 0x7B...0x7F)

For backward compatibility, the Administrator password quality check can be switched off through the SAC `pqAdminPQ` property.

Initialization Key password quality check cannot be switched off.

> **NOTE**   The password quality is in use only when the Administrator Password and Initialization Keys are used in a 'Friendly' (textual) format. For more information, refer to the 'Friendly Admin Password' section in the *SafeNet Authentication Client User Guide*.
>
> eToken 5110 FIPS and eToken 5110 devices support only *Friendly Admin* passwords.
>
> If a customer does not want to be compliant with these PIN Quality policies, use hexadecimal keys (also through SAC UI and SAC API). Friendly Admin PIN length can be 24 binary or 48 hexadecimal. The Initialization Key length can be 32 binary or 64 hexadecimal. In this case, the keys are used as-is (without derivation) and PIN Quality is not checked.

SAC supports password quality settings for the User PIN. The implementation of these settings may differ on various devices. User PIN policies are created or modified during a device's initialization process or during the device's life cycle after Administrator (SO) authentication.

Depending on the device model (for example: IDPrime or eToken devices) and initialization mode that is set (for example: the device is initialized without password policies), password quality policies are enforced by the device or by the middleware software (SAC).

Device Type	Where the policy is stored:	Policy is enforced by:
eToken 5110 GA eToken 5110 FIPS	Depends on how the device is formatted: On board SAC configuration	Middleware
IDPrime MD 840/3840 SafeNet IDPrime 940/3940 eToken 5110 CC	On board	Middleware (except for the PIN length, which is validated on board)
IDPrime MD 830/3811 SafeNet IDPrime 930/3930 eToken 5300	On board	On board

> **NOTE**   Each device (IDPrime / eToken) has a different policy setting. For more information, refer to the Token Settings chapter in *SafeNet Authentication Client User Guide*.

The SAC *Client Settings* policy is currently used only on eToken 5110 GA and 5110 FIPS. This policy is used in the following cases:

> The device is initialized without on board policies

> The default values used during the device initialization flow

PIN Retry Counter

Setting the Administrator/User PIN Retry Counter may vary depending on your device type:

Administrator PIN Retry Counter

> **SafeNet IDPrime MD 840** - The Administrator PIN Retry Counter cannot be modified on this device.

> **SafeNet IDPrime 940/3940** - The Administrator PIN Retry Counter is supported. The parameter is configured during factory settings and therefore, cannot be modified.

> **SafeNet IDPrime MD 830 B / SafeNet IDPrime 930/3930** - The Administrator PIN Retry Counter is supported. The parameter can be modified using SAC.

User PIN Retry Counter

SafeNet eToken 5110 FIPS or SafeNet eToken 5110 - Due to an eToken applet limitation, the User Retry Counter cannot be set on these smart cards, unless they are initialized.

Collecting SAC Logs

Collecting SAC logs allow administrators and technical-support personnel to diagnose the source of many problems that may have occurred while working with SAC. This information is used for debugging purposes.

SAC logs are collected by the following method:

> SAC GUI (SAC Tools)

> SAC Core (No UI)

To Enable SAC Logs through SAC GUI (SAC Tools)

Perform the following steps:

1. Open **SAC Tools** > **Advanced View** > **Client Settings**, and click the **Advanced** tab.

2. Click **Enable Logging**.

   The button will change to: Disable Logging. (For more information, refer to 'Enable Logging' in *SafeNet Authentication Client User Guide*.

3. Restart the application that requires the debug logs to be created.

> **NOTE** SAC Log files are created in the following directory `/tmp/eToken.log`.

To Enable SAC Logs through SAC Core (No UI)

Perform the following step:

1. Edit the `/etc/eToken.common.conf` file and type as below:

    ```
    [Log]
    Enabled = 1
    ```

    Where:

    •   0: Logs collection is disabled

    •   1: Logs collection is enabled

CHAPTER 2: Common Criteria

The IDPrime Applets 4.0, 4.2, 4.4 are Common Criteria certified on Common Criteria based smart cards and tokens. . These devices can have certain parameters customized in the factory with values that differ from the default profile. For a detailed list of supported cards, refer to *SafeNet Authentication Client Release Notes*

> **NOTE** The IDPrime MD 840/ 3840 cards or eToken 5110 CC do not support modifying the retry counter on the Admin Key. The recommended workaround is to set the profiles with a PUK instead of the Admin Key.
> To ensure maximum security, when using friendly mode, set the password with at least 16 random printable characters.

The following parameters can be customized:

> Number and type of key containers

> Support of RSA 4,096-bit key containers

> PINs (#1, #3 and #4 only)

> Try Limit

> Unblock PIN (PIN#1 only)

> PIN validity period

> Secure messaging in contactless mode

Number and Type of Key Containers

The following are the default settings. For other options consult your Thales representative.

By default, the IDPrime Applet 4.0 is pre-personalized with:

> 2 X 2,048-bit CC Sign Only RSA Keys

> 2 X 1,024-bit Standard Sign and Decrypt RSA Keys

> 8 X 2,048-bit Standard Sign and Decrypt RSA Keys

> 2 X 256-bit Standard Sign and Decrypt EC Keys

By default, the IDPrime Applet 4.4.2 is pre-personalized with:

> 2 X 2048-bit CC Sign Only RSA Keys

> 2 X 4096-bit CC Sign Only RSA Keys

> 2 X 256-bit CC Sign Only ECC Keys

> 8 X 2048-bit CC Sign and Decrypt RSA Keys

> 2 X 1024-bit CC Sign and Decrypt RSA Keys
> 2 X 4096-bit CC Sign and Decrypt RSA Keys
> 2 X 1024-bit CC Sign and Decrypt ECC Keys

> **NOTE**   The Key Generation method for Common Criteria key containers is either OBKG or Key import.

Common Criteria API Adjustments

Below table provides a high-level description of the adjustments that are made to the Standard and Extended PKCS#11 API to work with IDPrime CC devices. For more detailed information, refer to the code samples.

Standard PKCS#11 API	Extended PKCS#11 API
The `C_InitToken` function must receive the current Security Officer (SO) Password	The `C_InitToken` function must receive the current Security Officer (SO) Password
> When the `C_InitToken` function is called, you can enable link mode on the IDPrime CC device. Refer to " LinkMode " on page 46. > To revert a device back to unlinked mode after it was initialized in linked mode, use the PKCS#11 Extended API, or by using SAC Tools initialization process.	> To initialize the IDPrime CC device, the `ETCKA_CC` attribute must be set to `CK_TRUE`. > To initialize a device in linked mode, set the `ETCKA_IDP_CC_LINK` attribute to `1`. > To pass the current Digital Signature PUK value, use the `ETCKA_IDP_CURRENT_PUK` attribute. > To revert a device back to unlinked mode after it was initialized in linked mode, set the `ETCKA_IDP_CC_LINK` attribute to 0 and use the `ETCKA_PUK` attribute to set the new Digital Signature PUK value.
If a device is not configured to use linked mode, the `C_InitToken` function ignores the Digital Signature PUK and Digital Signature PIN.	If a device is not configured to use linked mode, use the `ETCKA_PUK` attribute to set the new Digital Signature PUK value.
> After the device has been initialized in linked mode, the `C_InitPIN` function initializes the Digital Signature PIN and the User PIN. Both PIN's are set to the same value. > The `C_SetPIN` function used with the `CKU_SO` flag changes both the Administrator PIN and Digital Signature PUK to a new value. For details on Friendly Admin Password, refer to *SafeNet Authentication Client User Guide*. > The `C_InitPIN` function used with the `CKU_USER` flag changes both the User PIN and Digital Signature PIN to a new value.	If the device is initialized to use linked mode, the `C_InitPIN` function and `C_SetPIN` function behaves the same as described in the *Standard PKCS#11* section.

CHAPTER 3:   Installation

This chapter provides the installation procedures for SafeNet Authentication Client (SAC) 10.8 R2 (GA) Mac. Local administrator rights are required to install or uninstall it.

> **NOTE** If IDGo 800 PKCS#11 is installed, be sure to remove it before installing this release.

Installation Files

The software package provided with this release includes the following files and documentation for install/upgrade.

File	Description
Installation File	
`SafeNetAuthenticationClient.10.8.xx.dmg`	> Installs SafeNet Authentication Client with UI. The .dmg disk image contains SAC with UI install and uninstall applications. > Installs/Uninstalls SafeNet Authentication Client with UI.
`SafeNetAuthenticationClient.10.8.xx.Core.dmg`	> Installs SafeNet Authentication Client without UI. The .dmg disk image contains SAC without UI install and uninstall applications. > Installs/Uninstalls SafeNet Authentication Client without UI.
Documentation Files	
`007-013724-004_SafeNet Authentication Client_ 10.8_R2_Mac_GA_Release_Notes_Rev.C`	SafeNet Authentication Client 10.8 R2 (GA) Release Notes for Mac. Read before installation for last minute updates that may affect installation; contains important information such as resolved and known issues and troubleshooting for Mac.

File	Description
`007-013725-004_SafeNet Authentication Client_ 10.8_R2_Mac_GA_User_Guide_Rev.C`	SafeNet Authentication Client 10.8 R2 (GA) User Guide for Mac. Provides detailed information for the user and system administrator regarding the use of SAC for Mac.
`007-013726-004_SafeNet Authentication Client_ 10.8_R2_Mac_GA_Administrator_Guide_Rev.C`	SafeNet Authentication Client 10.8 R2 (GA) Administrator Guide for Mac (this document). Provides detailed information for the system administrator regarding the installation, configuration, maintenance, and management of SAC for Mac.

This section describes the different files/data (compiled programs, images, media, and compressed files) that are saved in various directories after SAC Mac is installed.

SAC Mac installation is available as **SAC Standard** (`SafeNetAuthenticationClient.10.8.xx.dmg`)

SAC Standard Installation Files

The SAC standard installation includes the following:

> PKCS#11 for all supported cards

> SafeNet Authentication Client (Tray icon) application

> SAC Tools

> CTK plug-ins

SAC provides the following PKCS#11 API for third-party application integrations. The following symbolic links are added to the `/usr/local/lib/pkcs11` folder:

File	Description
`libeTpkcs11.dylib`	Symbolic link to libeToken.dylib (`/Library/Frameworks/eToken.framework/Versions/Current/`)
`libIDPrimePKCS11.dylib`	Symbolic link to libIDPrimePKCS11.dylib (`/Library/Frameworks/eToken.framework/Versions/Current/`)
`libClassicClientPKCS11.dylib`	Symbolic link to libClassicClientPKCS11.dylib (`/Library/Frameworks/eToken.framework/Versions/Current/libClassicClientPKCS11.dylib`)

The following symbolic links and `SafeNet Extension.app` are installed in the `/Applications/SafeNet/SafeNet Authentication Client` folder:

File	Description
`SafeNet Authentication Client Tools.app`	Symbolic link to SafeNet Authentication Client Tool application (`/Library/Frameworks/eToken.framework/Versions/A/`) **NOTE** Applicable for SAC with UI only.
`SafeNet Authentication Client.app`	Symbolic link to SafeNet Authentication Client application (`/Library/Frameworks/eToken.framework/Versions/A/`) **NOTE** Applicable for SAC with UI only.
`SafeNet Extension.app`	Crypto Token Kit (CTK) plugin bundle is part of `SafeNet Extension.app` application. It can be found in `SafeNet Extension.app/Contents/PlugIns/PKCS11 Token.appex` **NOTE** Applicable for SAC with and without UI.

The following files are installed in the `/etc` folder:

File	Description
`eToken.conf`	Used to configure SAC properties. Refer to "Configuration Properties" on page 37.
`eToken.common.conf`	

All SAC files are installed in the `/Library/Frameworks/eToken.framework/` folder:

File	Description
`SafeNet Authentication Client.app/`	SafeNet Authentication Client (Tray icon) application. **NOTE** Applicable for SAC with UI only.
`SACTools`	SafeNet Authentication Client Tools application. **NOTE** Applicable for SAC with UI only.
`SACSrv`	SafeNet Authentication Client Service application. **NOTE** Applicable for SAC with UI only.
`libeToken.dylib`	Cryptography module (mandatory for smart card support)

File	Description
`libIDPrimePKCS11.dylib`	IDPrime PKCS#11 module
`libClassicClientPKCS11.dylib`	Classic Client PKCS#11 module for IDClassic 340 (V3) cards. (Read only functionality)
`libSACUI.dylib`	SAC UI library. **NOTE** Applicable for SAC with UI only.
`libSACLog.dylib`	SAC log feature
`libIDPrimeTokenEngine.dylib`	The IDPrime token/card engine
`libClassicClientTokenEngine.dylib`	Support for IDClassic cards engine
`SACHelp.pdf`	SafeNet Authentication Client User Guide. **NOTE** Applicable for SAC with UI only.
`libcrypto.1.1.dylib`	OpenSSL 1.1 library

The following symbolic links are added to the `/usr/local/lib` folder:

File	Description
`libIDPrimePKCS11.dylib`	Symbolic link to libIDPrimePKCS11.dylib (`/Library/Frameworks/eToken.framework/Versions/Current/`)
`libeTpkcs11.dylib`	Symbolic link to libeToken.dylib (`/Library/Frameworks/eToken.framework/Versions/Current/`)
`libSACLog.dylib`	Symbolic link to libSACLog.dylib (`/Library/Frameworks/eToken.framework/Versions/Current/`)
`libSACUI.dylib`	Symbolic link to libSACUI.dylib (`/Library/Frameworks/eToken.framework/Versions/Current/`) **NOTE** Applicable for SAC with UI only.
`libeToken.dylib`	Symbolic link to (`/Library/Frameworks/eToken.framework/Versions/Current/`)

Installing SACwith/without UI on macOS

Use the SAC installation wizard to install the application with its default properties and features.

> **NOTE** No UI components are available if SAC without UI is installed on macOS.
> The SAC without UI installer package installs all the required libraries except any UI
> component.

Installation Steps

Perform the following steps to install SAC with UI and without UI on macOS:

1. Double-click the following file available in the installation package based on your requirement:

   - **For SAC with UI-** `SafeNetAuthenticationClient.10.8.x.x.dmg`

   - **For SAC without UI-** `SafeNetAuthenticationClient.10.8.xx.Core.dmg`

   A new disk image file is created in the **Finder** window, including a package installation file and an uninstall application.
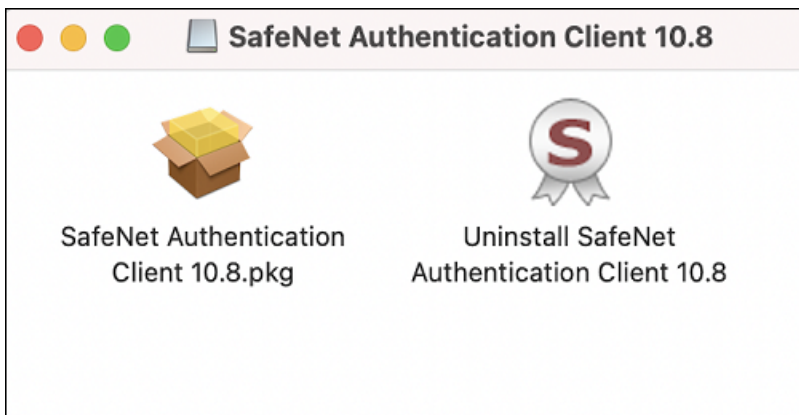
   The above window is displayed if SAC with UI file is selected for installation.

   The above window is displayed if SAC without UI file is selected for installation.

> **NOTE** Step 2 to 4 are common for both (SAC UI and SAC without UI).

2. Double-click **SafeNet Authentication Client 10.8.pkg** to start the installation.

   The **Welcome to the SafeNet Authentication Client Installer** window is displayed.
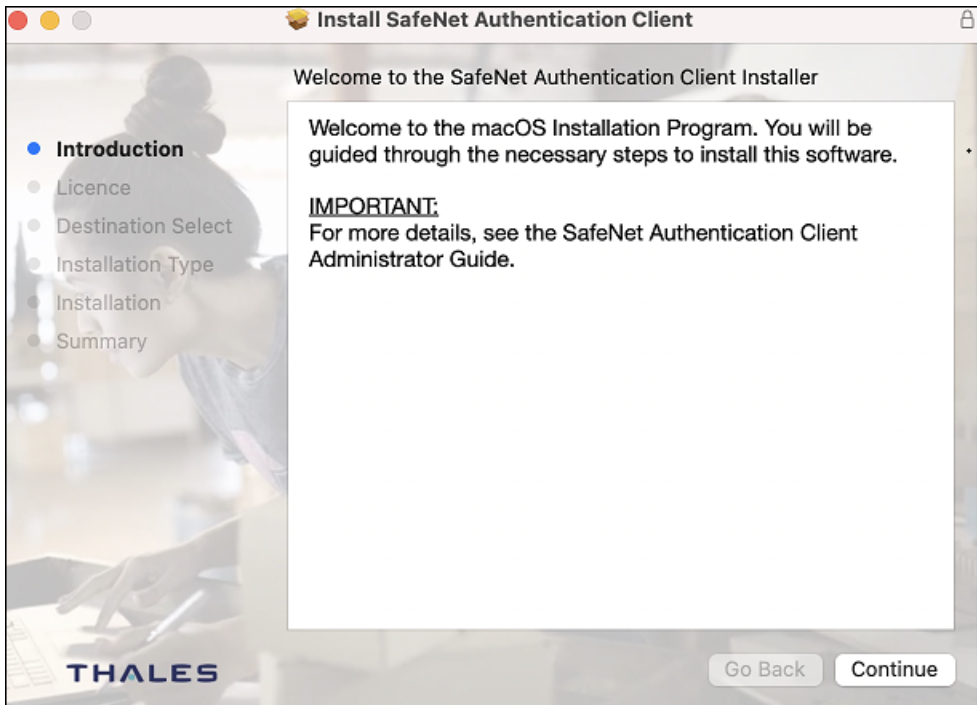
3. Click **Continue**.

   The **Software License Agreement** window is displayed.
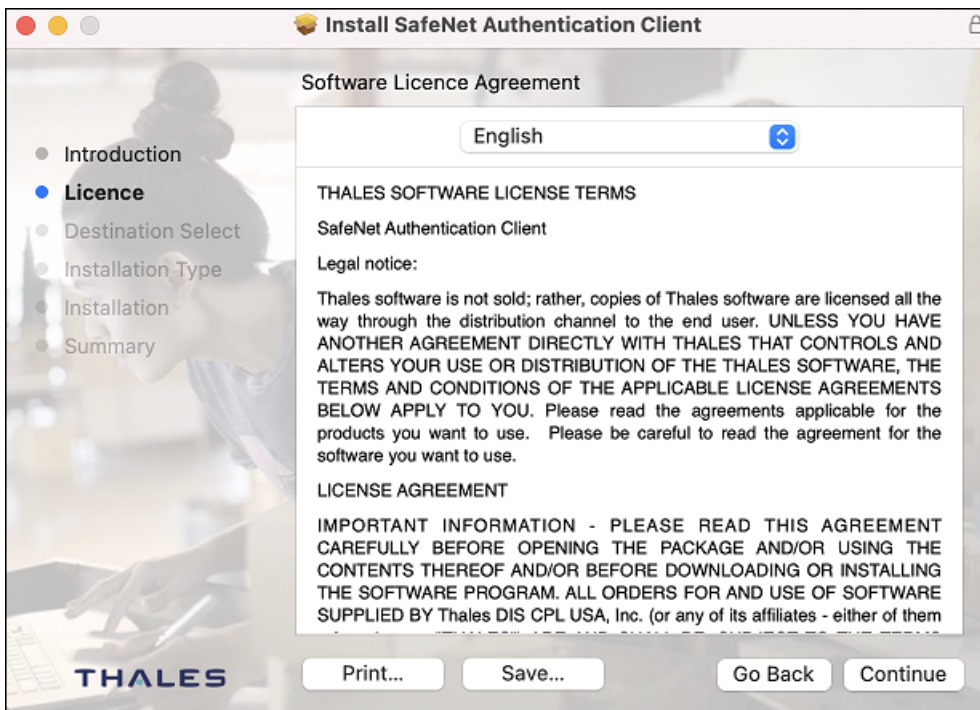
4. Click **Continue**.

   The **Agreement** pop-up appears.
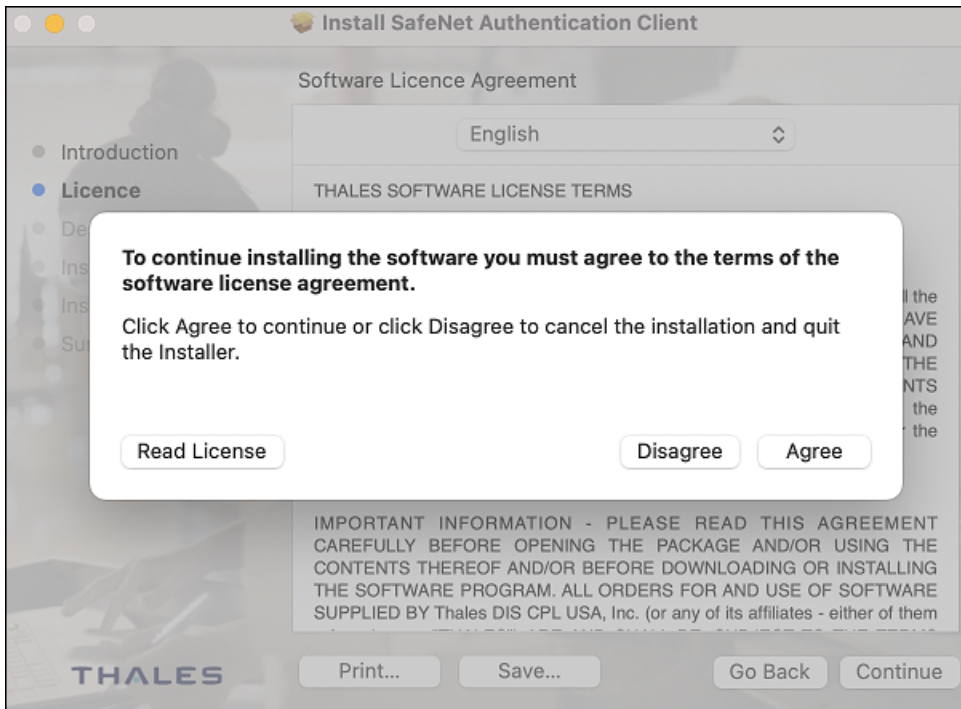
5. Click **Agree** to accept the software license agreement.

The **Standard Install** window is displayed.
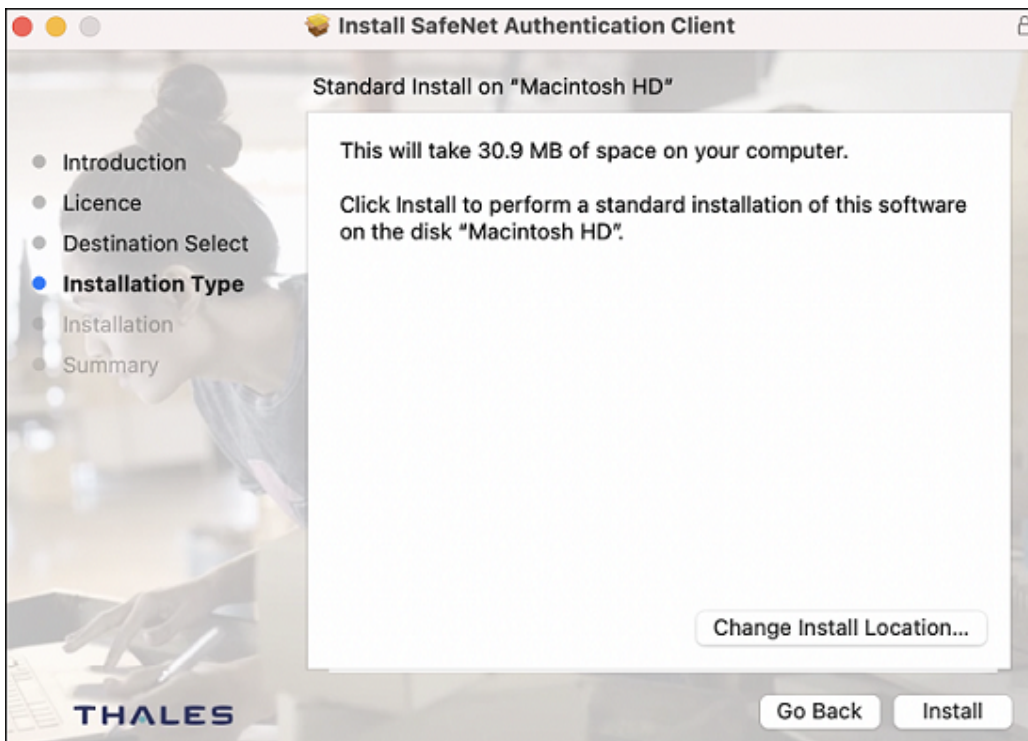
The above window is displayed for SAC with UI.
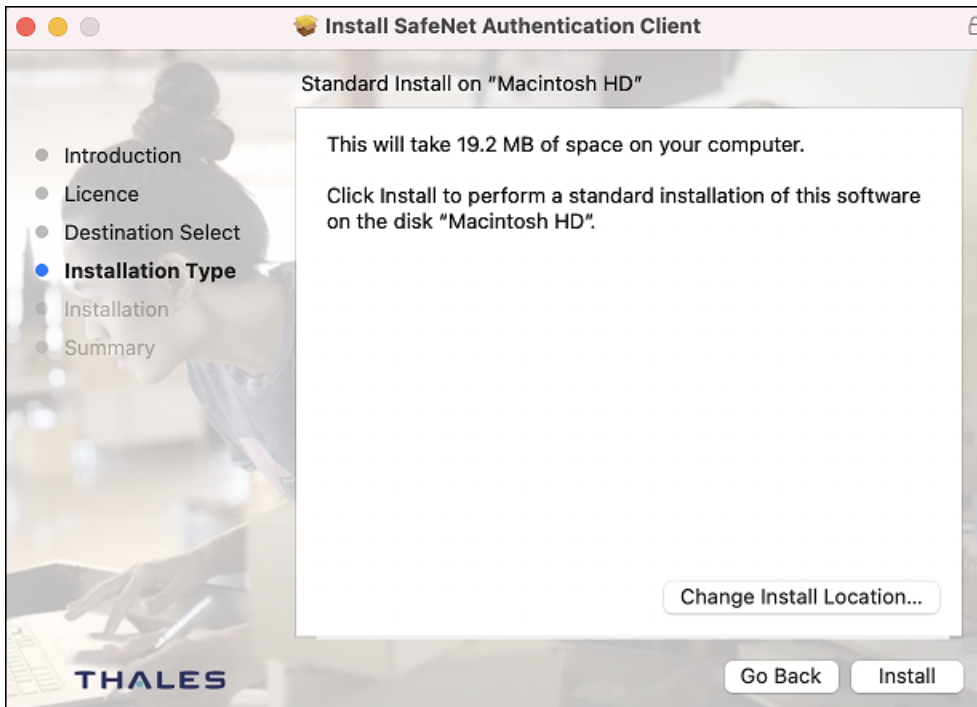
The above window is displayed for SAC without UI.

> **NOTE** Step 6 to 9 are common for both (SAC UI and SAC without UI).

6. Click **Install**.

   The **Authenticate** window is displayed.

7. Enter username and password, then click **Install Software**.

> **NOTE** Administrator permissions are required to install SAC.

SafeNet Authentication Client is installed successfully, and below screen is displayed.
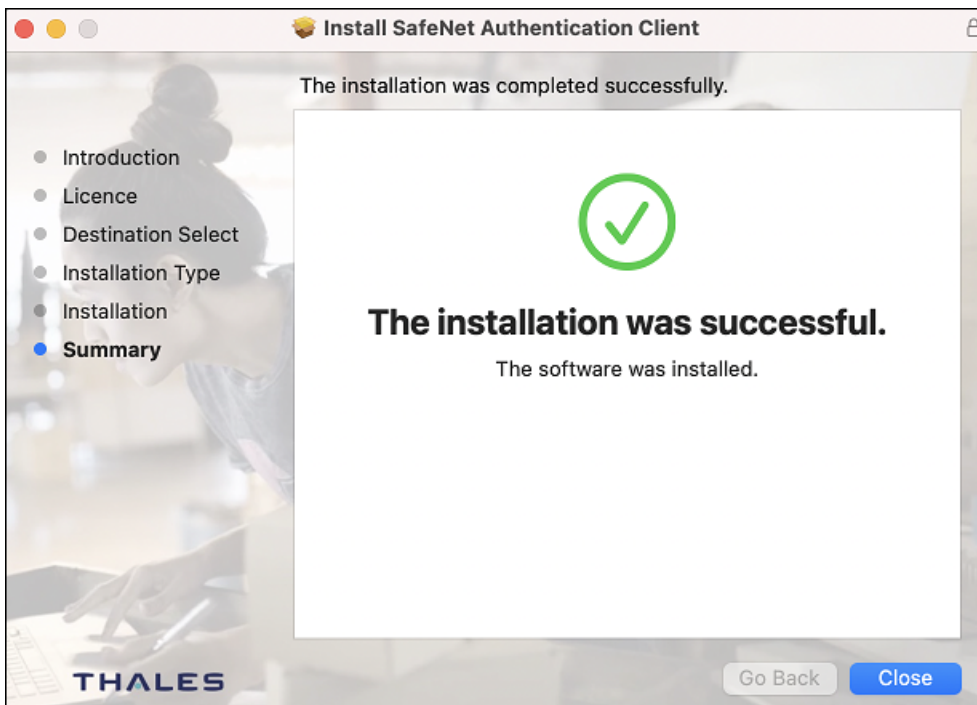
8. Click **Close**, and perform a **Restart** (recommended).

9. Log in again to macOS.

Installing SAC from the Mac Terminal

> **NOTE** Below steps are applicable for both (SAC UI and SAC without UI).

Perform the following steps to install SAC from the Mac terminal:

1. Extract the `SafeNet Authentication Client 10.8.pkg` file from the `dmg` file.

2. Run following command at the location in the terminal where the file is extracted:
   `sudo installer -pkg./SafeNet\Authentication\Client\ 10.8.pkg/ -target /`

3. Enter root password when prompted. The SafeNet Authentication Client is installed.

4. Restart macOS (recommended).

Upgrading SAC on a Mac

It is recommended to upgrade the SafeNet Authentication ClientSAC to the latest version on each computer that uses a SafeNet eToken, or SafeNet smart card. Local administrator rights are required to upgrade SafeNet Authentication ClientSAC.

After upgrading from SafeNet Authentication ClientSAC 10.8 R1 to SAC 10.8 R21 on a Mac, it is recommended that you restart the machine in order to recognize the device.

Loading the Token PKCS#11 Security Module

The PKCS#11 security module must be loaded to use SafeNet Authentication Client.

> **NOTE**
> - Ensure that there is only one loaded security module having a path with the value: `libeTPkcs11.dylib`
> - For information on how to work with Multi-Slots, refer to the PKCS#11 Digital Signature PIN Authentication section of *SafeNet Authentication Client User Guide*.

Locations of PKCS#11 Security Module

SAC PKCS#11 library files are installed in `/usr/local/lib/`. This location must be updated in applications using the SAC PKCS#11 module, such as Mozilla Firefox, Thunderbird, or Adobe Reader.

Configuring Acrobat Security Settings

Adobe Acrobat can be configured to protect PDF documents using a `.CER` certificate.

To Set the Adobe Acrobat Security Settings

Perform the following steps:

1. Open Adobe Acrobat and select **Preferences** > **Signatures** > **Identities & Trusted Certificates** > **More**.

   The **Digital ID and Trusted Certificate Settings** window is displayed.

2. From the left panel, click **Digital IDs** > **PKCS#11 Modules and Tokens**.

3. If a PKCS#11 Module is not attached, click **Attach Module**, browse to SAC PKCS11 library `/usr/local/lib/libeTPkcs11.dylib`, and click **Open**.

   The connected token and certificate appear under PKCS#11 Modules and Tokens.

   > **NOTE**   For information on how to work with Multi-Slots, refer to the PKCS#11 Digital Signature PIN Authentication section of *SafeNet Authentication Client User Guide*.

To Digitally Sign a PDF using Adobe Acrobat

Perform the following steps:

1. Select **Tools** > **Certificates** > **Digitally Sign**.

   A pop-up is displayed asking to select the area where the signature appears.

2. Click **OK** and select the area where you want the signature to appear.

   The **Sign with a Digital ID** window is displayed.

3. Select the Digital ID that you want to use for signing and click **Continue**.

   The **Save As** window is displayed.

4. Do the following and click **Save**:

   • In the **Save As** field, enter the name of the certificate.

   • From the drop-down or left panel, select the location where you want the certificate to be saved.

   • From the **Format** drop-down, select the format in which the certificate to be saved.

5. A pop-up appears to authenticate the user, enter your token **PIN** and click **OK**.

   The digital signature is added successfully at the required location in the pdf.

Configuring Mozilla Firefox\Thunderbird

Perform the following steps:

1. Do one of the following:

   • When working with Firefox, go to **Firefox** > **Preferences** > **Privacy & Security** > **Certificates** > **Security Devices**.

   • When working with Thunderbird, go to **Thunderbird** > **Preferences** > **Privacy & Security** > **Certificates** > **Security Devices**.

   The **Device Manager** window is displayed.

2. If eToken is not listed in the **Security Modules and Devices** column, click **Load**.

   The **Load PKCS#11 Device** window is displayed.
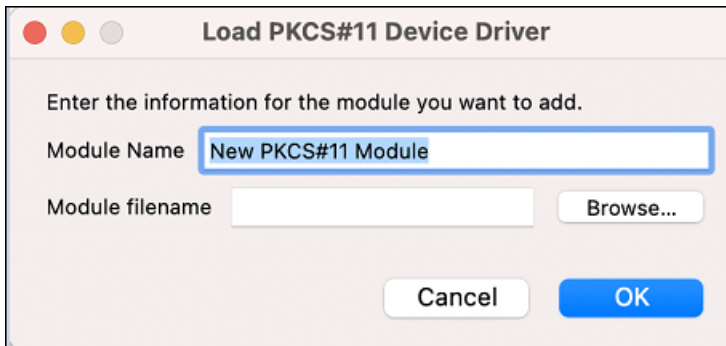
3. Replace the contents of **Module Name** with eToken.

4. In **Module filename**, enter `/usr/local/lib/libeTPkcs11.dylib`

> **NOTE** The *Module* fields are case sensitive.

> **NOTE**
> - To work with CC devices in unlinked mode, enter the following string for Multi-Slot support:
> `/usr/local/lib/libIDPrimePKCS11.dylib`
> - For information on how to work with Multi-Slots, refer to the PKCS#11 Digital Signature PIN
> Authentication section of *SafeNet Authentication Client User Guide*.
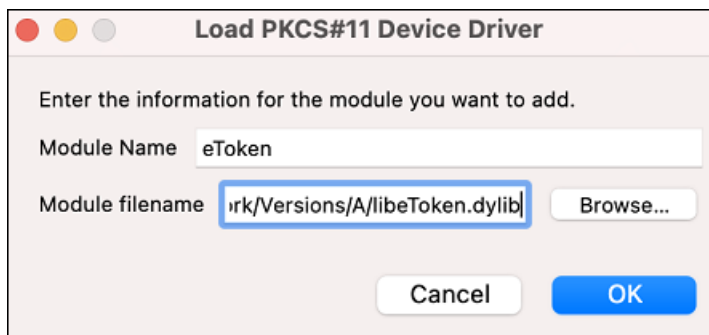
5. Click **OK**.

eToken is listed in the **Security Modules and Devices** column of the **Device Manager** window.
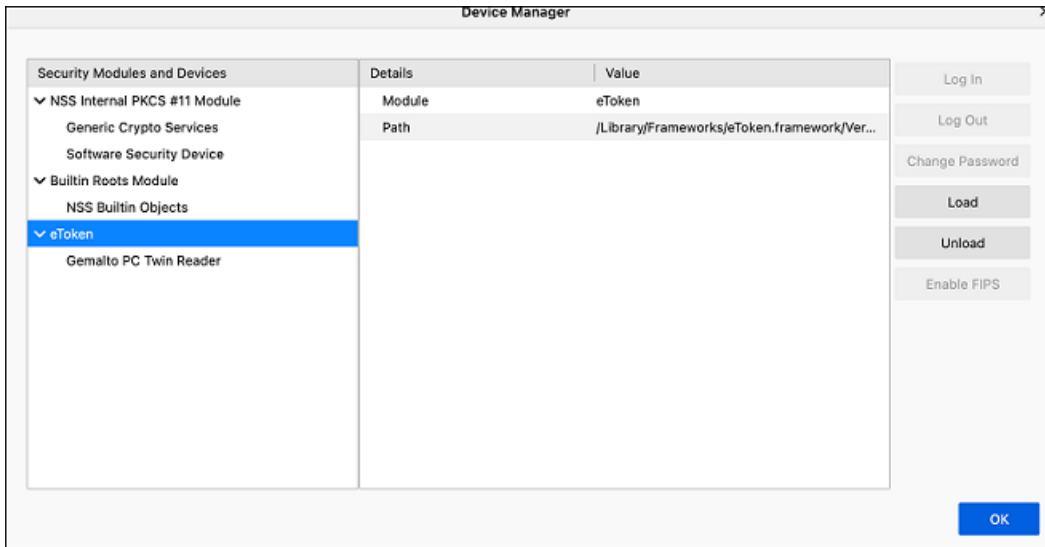
6. Click **OK** to exit the **Device Manager**.

7. Restart Firefox\ Thunderbird.

Installing CCID Driver for SafeNet eToken Fusion CC tokens

> **NOTE**
> > On macOS Monterey v12.x, the CCID driver v1.4.36 installation is needed to add support for SafeNet eToken 5110+FIPS and SafeNet eToken 5300-C.
> > On macOS Monterey and Ventura, CCID driver v1.5.1 is needed to add support for eToken Fusion CC tokens.

> **NOTE**

Perform the following steps to install CCID driver:

1. Log on as an administrator.

2. Close all applications.

3. Double-click the **ccid-Installer.dmg** available on the support portal.

   A new disk image file is created in the **Finder** window, including a package of ccid installer.
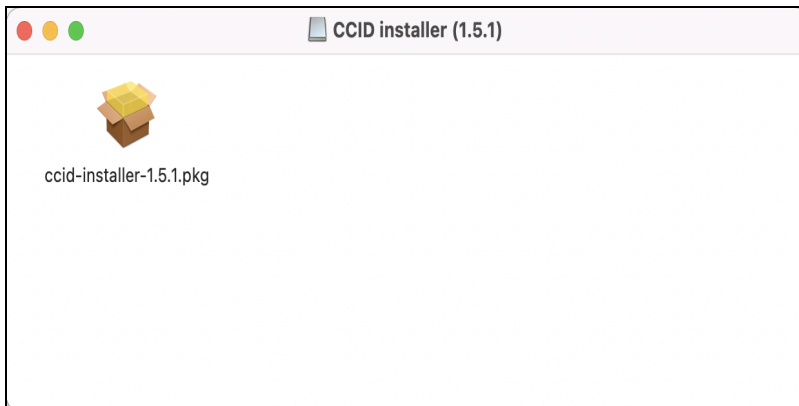
4. Double-click **ccid-installer-1.5.1.pkg** to start the installation.

   The **Welcome to the CCID driver for SafeNet eTokens Installer** window is displayed.
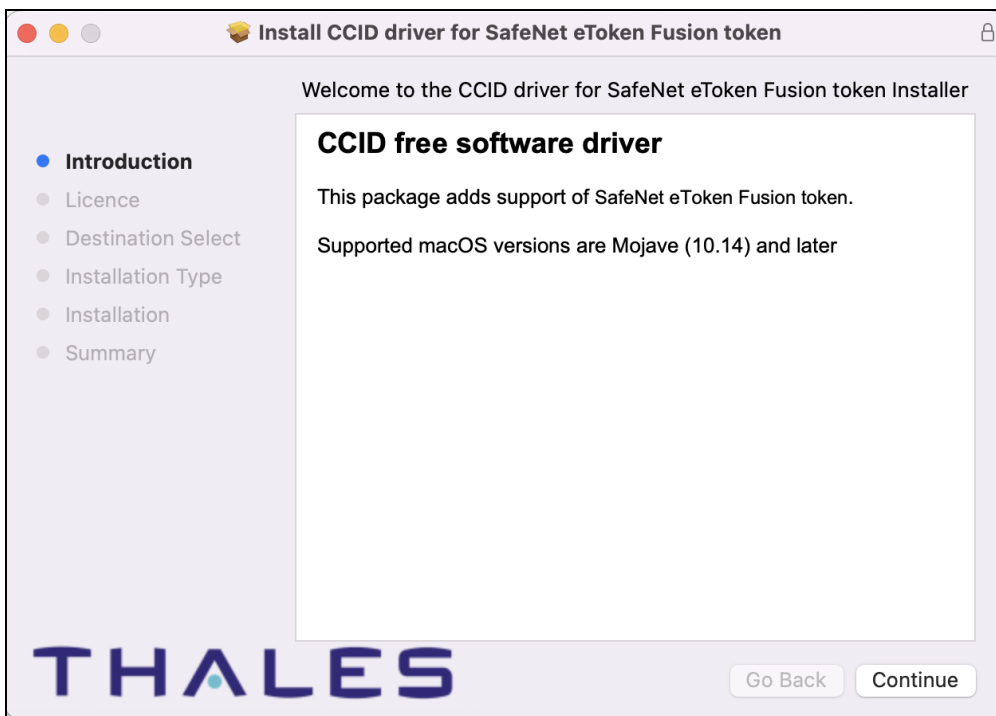
5. Click **Continue**.

   The **Software License Agreement** window is displayed.
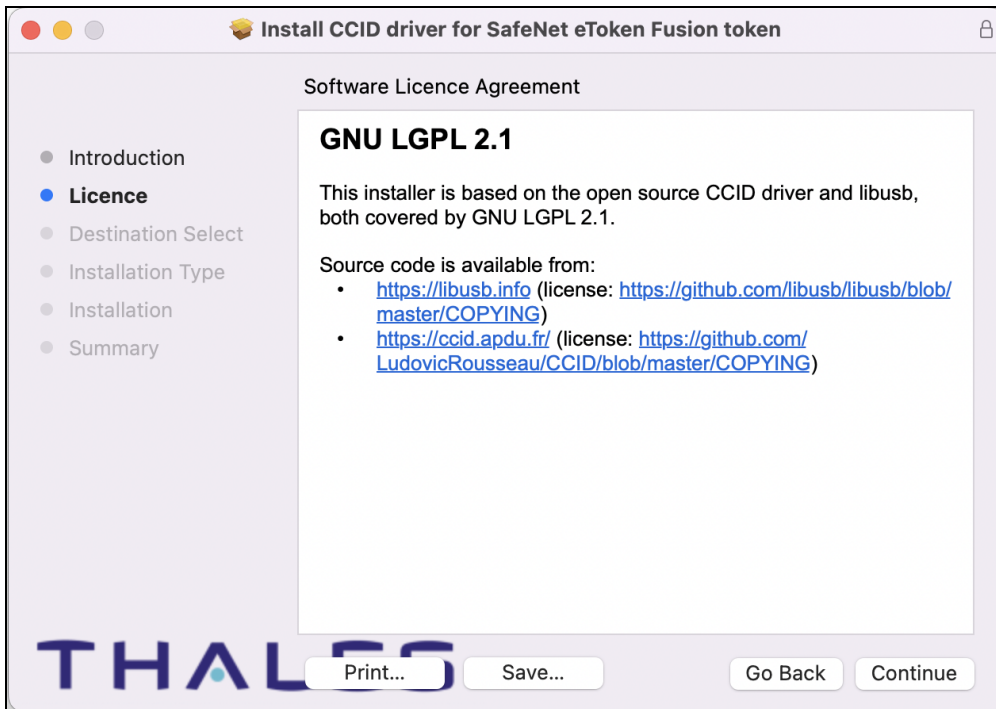
6. Click **Continue**.

The **Agreement** pop-up appears.
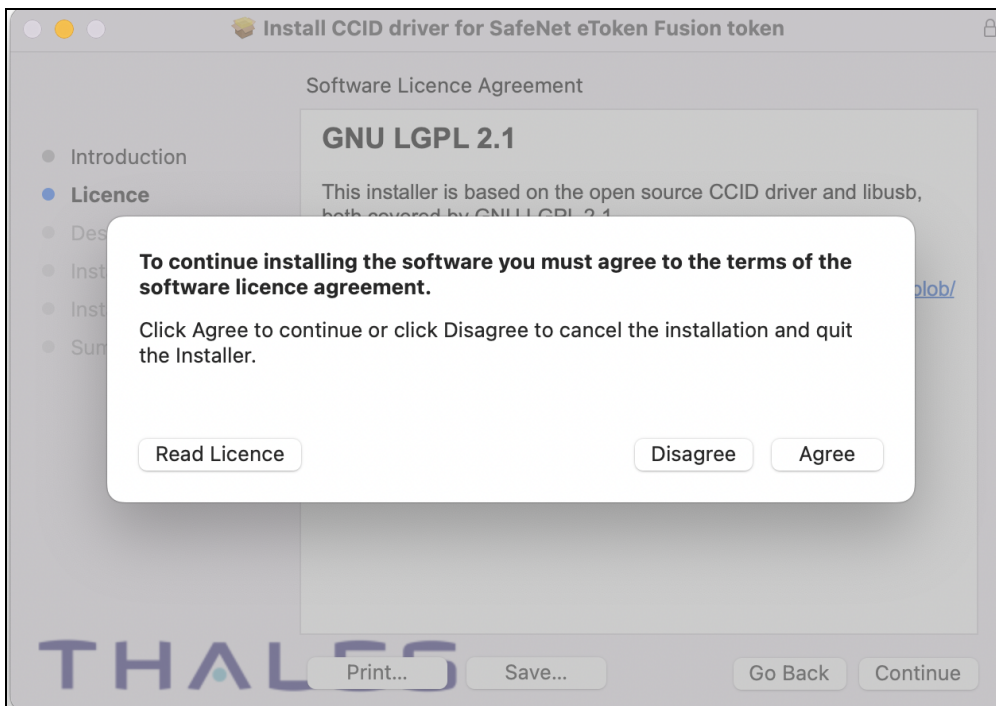
7. Click **Agree** to accept the software license agreement.

The **Standard Install** window is displayed.
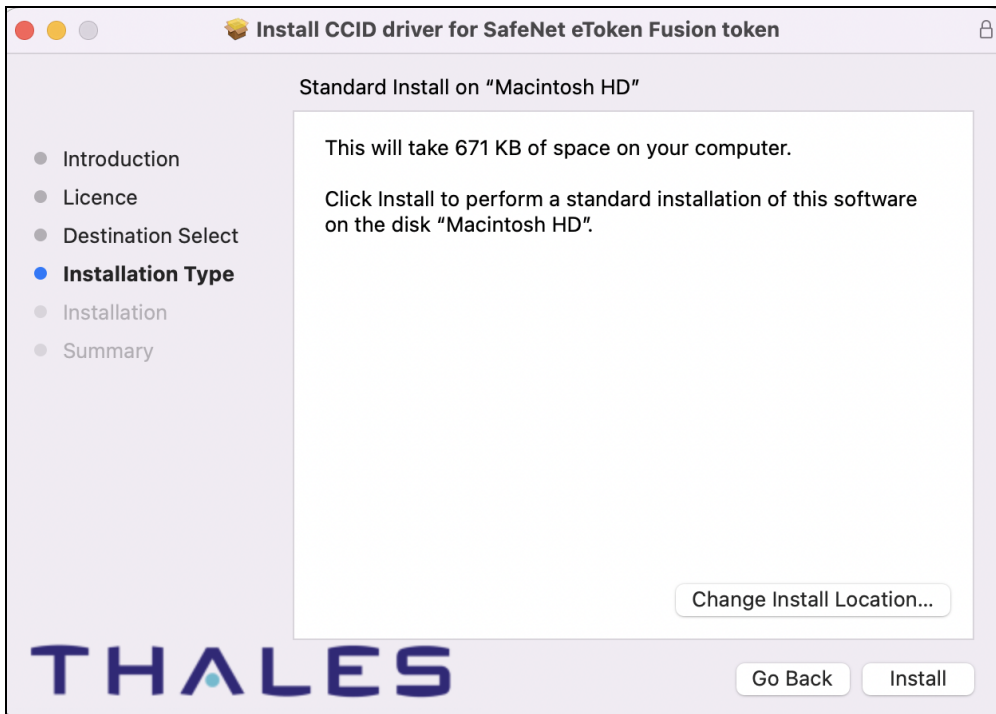
8. Click **Install**.

The **Authenticate** window is displayed.

9. Enter username and password, then click **Install Software**.

> **NOTE** Administrator permissions are required to install the driver.

CCID driver for SafeNet eToken is installed successfully, and below screen is displayed.
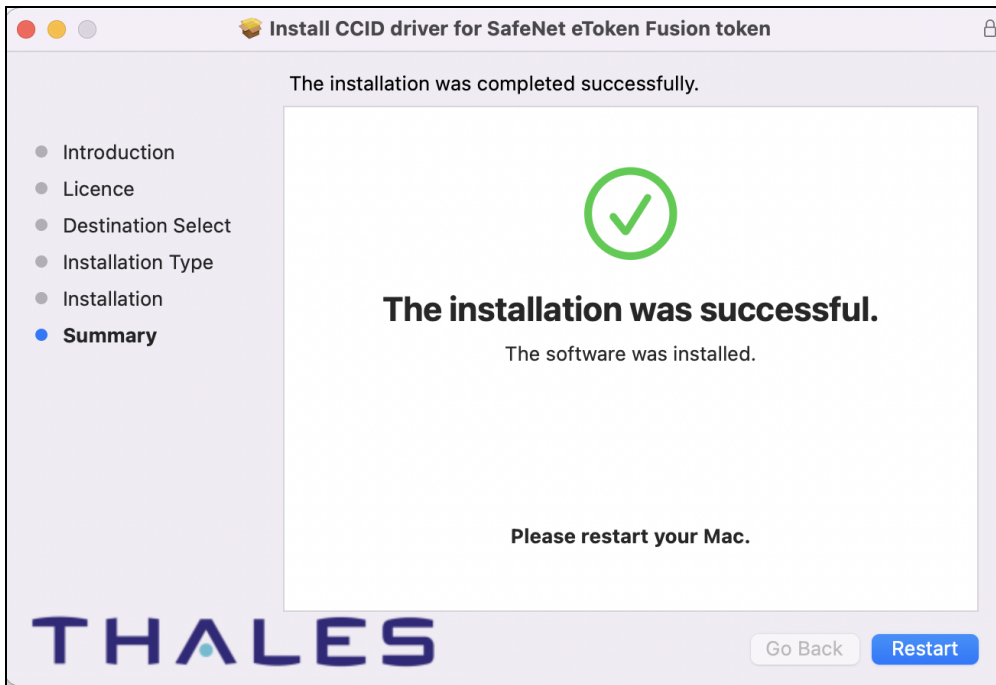
10. Click **Close**, and perform a **Restart**.

CHAPTER 4:   Crypto Token Kit Modules

SafeNet Authentication Client (SAC) 10.8 R2 (GA) Mac supports Crypto Token Kit (CTK) module.

On macOS 10.12 and above, users can create NSExtension-based smart card drivers that allow the contents of certain smart cards to be present as part of the system keychain. This mechanism replaces the deprecated Common Data Security Architecture.

The support for Mac new CTK enables accessing Smart Cards and manages user interactions. The new CTK module provides programmatic access to Smart Cards.

To check if the CTK module is enabled or disabled, run the following command:

```
pluginkit -vv -m -p com.apple.ctk-tokens
```

When the CTK module is enabled, the following is displayed:

When the CTK module is disabled (Thales PKCS#11 information is not available), the following is displayed:

```
[users-Mac-mini-3:~ user$ pluginkit -vv -m -p com.apple.ctk-tokens
          com.apple.CryptoTokenKit.pivtoken(1.0)
                    Path = /System/Library/Frameworks/CryptoTokenKit.framework/PlugIns/pivtoken.appex
                    UUID = AE3A0C2F-3BA7-49CA-9A28-D149527A6A2D
               Timestamp = 2021-04-06 09:29:21 +0000
                     SDK = com.apple.ctk-tokens
            Display Name = Personal Identity Verification token driver
              Short Name = pivtoken


          com.apple.CryptoTokenKit.setoken(1.0)
                    Path = /System/Library/Frameworks/CryptoTokenKit.framework/PlugIns/setoken.appex
                    UUID = 44F3B50C-565C-4D92-94F8-FF1487C33E9D
               Timestamp = 2021-04-06 09:29:21 +0000
                     SDK = com.apple.ctk-tokens
            Display Name = Secure Enclave Private Key Storage
              Short Name = setoken


  (3 plug-ins)
users-Mac-mini-3:~ user$
```

CHAPTER 5: Uninstall

After SafeNet Authentication Client (SAC) 10.8 R2 (GA) Mac is installed, you can uninstall it. Local administrator rights are required for uninstall. When SAC is uninstalled, user configuration and policy files may be deleted.

> **NOTE** Before uninstalling this release, make sure SAC Tools is closed.

Perform the following steps to uninstall SAC with UI and without UI on macOS:

1. Double-click the following file based on your requirement:

   - **For SAC with UI-** `SafeNetAuthenticationClient.10.8.x.x.dmg`

   - **For SAC without UI-** `SafeNetAuthenticationClient.10.8.xx.Core.dmg`

   A new disk image file is created in the **Finder** window, including an `mpkg` installation file and an uninstall application.
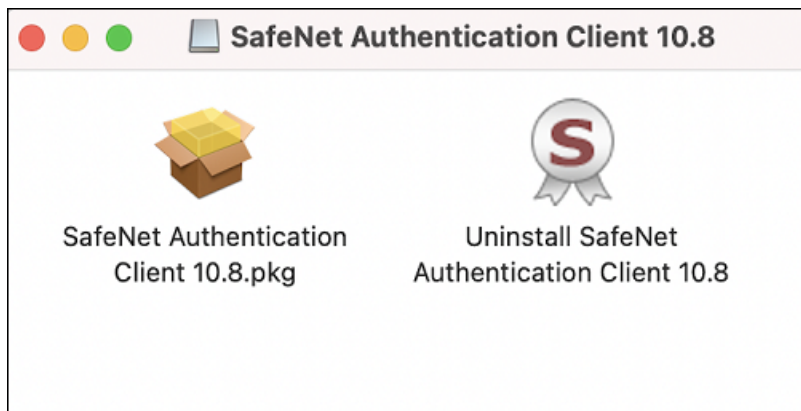
   The above window is displayed if SAC with UI file is selected for uninstallation.
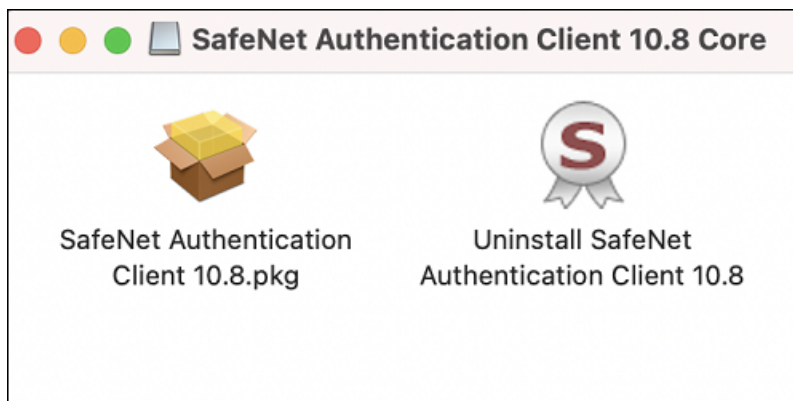
   The above window is displayed if SAC without UI file is selected for uninstallation.

> **NOTE** Step 2 to 5 are common for both (SAC UI and SAC without UI).

2. Click **Uninstall SafeNet Authentication Client 10.8**.

The **Welcome to the SafeNet Authentication Client Uninstaller** window is displayed.
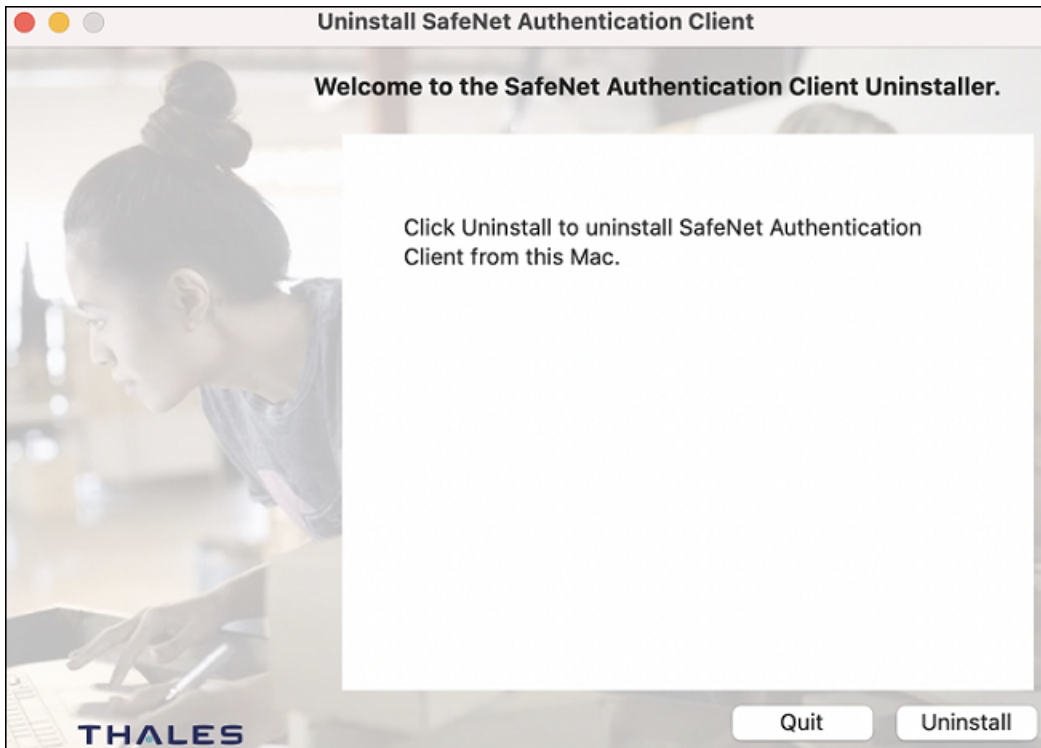
3. Click **Uninstall**.

The **Authenticate** window is displayed.
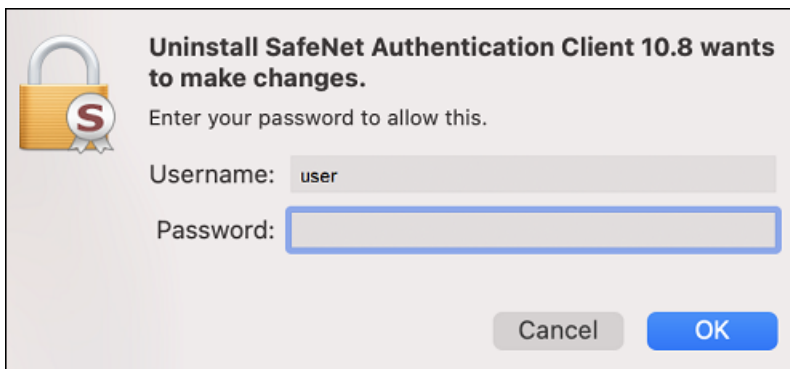
4. Enter the username, password, and click **OK**.

> **NOTE** You require administrator permissions to uninstall this release.

The uninstall completed successfully window is displayed.
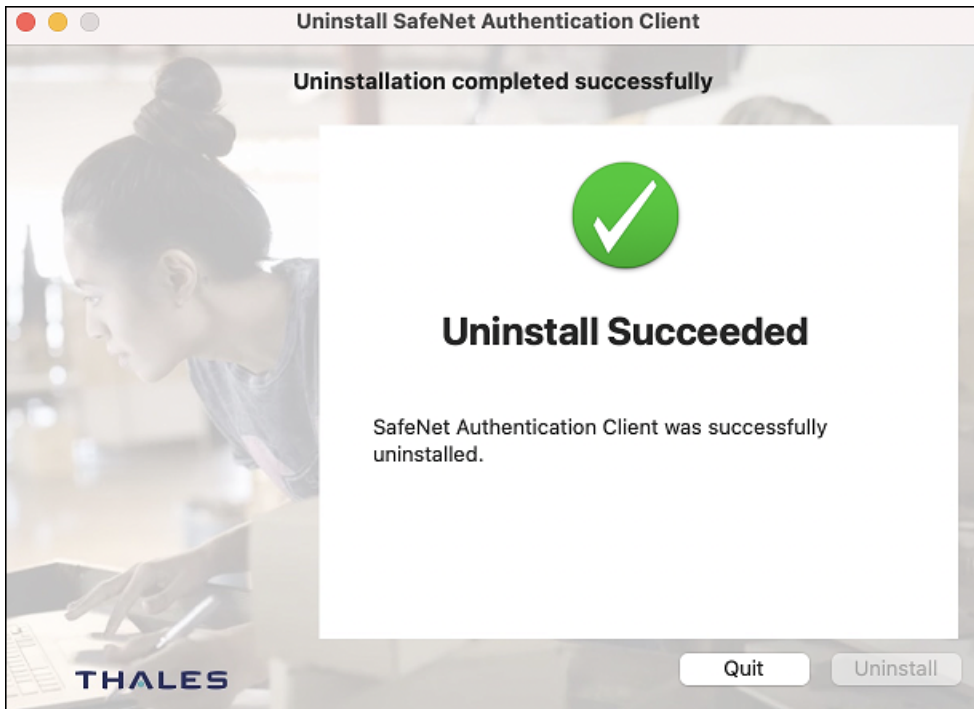
5. Click **Quit**.

CHAPTER 6:   Configuration Properties

SafeNet Authentication Client (SAC) properties are stored on the computer as `ini` files, which can be added and changed to determine SAC behavior. Depending on where an `ini` value is written, it applies globally, or limited to a specific user/application.

> **NOTE** All properties are set and edited manually.

SafeNet Authentication Client installs two configuration files:

> `/etc/eToken.conf`
> Requires administrator permissions(-rw-rw-r–)

> `/etc/eToken.common.conf`
> Does not require administrator permissions (-rw-rw-rw-)

Owner: root\admin

> **NOTE** eToken.common.conf contains settings for SafeNet eToken Virtual use onl

Configuration Files Hierarchy

To enable hierarchical priorities, up to three different versions of the eToken.conf configuration file can be created. For each key, the setting found in the file with highest priority determines the application's behavior. This design simulates the SafeNet Authentication Client (Windows) registry logic.

Windows Registry	Mac Installer	File Name	Priority	File Permissions
LM/Policies	Not provided	/etc/eToken.policy.conf	1(High)	Root
CU	Automatically created by GUI	~/.eToken.conf (located in user's home directory)	2	User
LM	Provided	/etc/eToken.conf	3	Root
LM	Provided	/etc/eToken.common.conf for eToken Virtual connections		All

> **NOTE**
> **>** `/etc/eToken.policy.conf` can be created manually by the system administrator.
> **>** Values defined in `/etc/eToken.policy.conf` are considered with highest priority and are in non editable mode in SAC Tools.

eToken Configuration Keys

All keys are located in `/etc/eToken.conf.`

General Settings

The following settings are written to the **General** section in the file `/etc/eToken.conf.`

> **NOTE**   On a macOS, the number of slots are determined by the PcscSlots and SoftwareSlots configuration keys described here. The *Reader Settings* window in SafeNet Authentication Client Mac Tools displays the number of slots that have been configured, but does not allow the user to change the settings.

Description	Value

Multi-Slot Support
 	
> Determines if SAC is backward compatible with Thales PKCS#11 Common Criteria devices (IDPrime MD 840, IDPrime MD 3840 and eToken 5110 CC).
 	
> The Mutli-Slot feature affects only SAC customized in compatible mode through `libIDPrimePKCS11.dylib`.
 	
Following are the two ways to work with IDPrime MD 840/940 or CC Cards, where a login is required for the Digital Signature Role:
 	
1. Use `libIDPrimePKCS11.dylib`, where the user has two smart cards:
  a. Physical Smart Card
  b. Virtual Smart Card (where Digital Signature Role is exposed as ROLE1 in the virtual smart card
 	
2. To enable the prompt login through a flag in the `/etc/eToken.conf` file, add the following line to Section [GENERAL]:
 	
`[GENERAL]`
 	
`EnablePrompt=1`
 	
This allows C_Login with Null or when a ROLE is not Logged in, a prompt is shown to enter the PIN/Password to complete the operation, such as C_SIGN / C_Encrypt/C_Decrypt ....
 	
For more information on Multi-Slots, refer to the PKCS#11 Digital Signature PIN Authentication section of *SafeNet Authentication Client User Guide*.
 	
> **NOTE** Linked Mode is not compatible with the Multi-Slot feature.
	Value Name: MultiSlotSupport
	
	Values: =0, =1
	> 1 - Multi-Slot support is enabled
	> 0 - Multi-Slot support is disabled
	
	Default: 1

PCSC Slots Defines the total number of PC/SC slots for all USB tokens and smart cards. Included in this total: > The number of allocated readers for third-party providers. > The number of allocated readers for other SafeNet physical tokens, which can be modified in *Reader Settings* in SafeNet Authentication Client Tools.	**Value Name:** PcscSlots **Values:** >=0 (0 = Physical tokens are disabled) **Default:** 8

Description	Value
Legacy Manufacturer Name > Determines if *Aladdin Knowledge Systems Ltd.* is written as the manufacturer name in token and token slot descriptions. > Use for legacy compatibility only.	**Value Name:** LegacyManufacturerName **Values:** > 1 - The legacy manufacturer name is written > 0 - The new manufacturer name is written **Default:** 0
Enable Private Cache > Determines if SafeNet Authentication Client allows the token's private data to be cached. > Applies only to tokens that were initialized with the private data cache setting. > The private data is cached in per process memory. > **NOTE** Can be set in SAC Tools.	**Value Name:** EnablePrvCache **Values:** > 1 (True) - Private data caching is enabled > 0 (False) - Private data caching is disabled **Default:** 1 (True)
Tolerate Finalize Determines if C_Finalize can be called by DllMain. > **NOTE** Define this property as per the process. Select this setting when using Novell Modular Authentication Service (NMAS) applications only.	**Value Name:** TolerantFinalize **Values:** > 1 (True) - C_Finalize can be called by DllMain > 0 (False) - C_Finalize cannot be called by DllMain **Default:** 0 (False)
Tolerate X509 Attributes Determines if CKA_SERIAL_NUMBER, CKA_SUBJECT, and CKA_ISSUER attributes can differ from those in CKA_VALUE during certificate creation > **NOTE** Enable TolerantX509Attributes when using certificates created in a non- DER encoded binary x.509 format. In some versions of PKI Client, this setting is not selected by default.	**Value Name:** TolerantX509Attributes **Values:** > 1 (True) - The attributes can differ > 0 (False) - Check that the values match **Default:** 0 (False)

Description	Value
Tolerate Find Templates Determines if PKCS#11 tolerates a *Find* function with an invalid template, returning an empty list instead of an error.	**Value Name:** TolerantFindObjects **Values:** > 1 (True) - A Find function with an invalid template is tolerated and returns an empty list > 0 (False) - A Find function with an invalid template is not tolerated and returns an error **Default:** 0 (False)
Protect Symmetric Keys Determines if symmetric keys are protected. > **NOTE** If selected, even non-sensitive symmetric keys are not extracted	**Value Name:** SensitiveSecret **Values:** > 1 - Symmetric keys cannot be extracted > 0 - Symmetric keys can be extracted **Default:** 0
Cache Marker Timeout Determines if SAC Service periodically inspects the cache markers of connected tokens for an indication that token content has changed. > **NOTE** If tokens are initialized as *eToken PKI Client 3.65 compatible* in SAC 8.0 and later, set this value to 0 to improve the performance.	**Value Name:** CacheMarkerTimeout **Values:** > 1 - Connected tokens' cache markers are periodically inspected > 0 - Connected tokens' cache markers are never inspected **Default:** 0
Override Non-Repudiation OIDs > Overrides SAC's list of standard certificate OIDs that require a high level of security. > **NOTE** Users must log on to their tokens whenever signing with a certificate defined as non-repudiation. > Avoid authenticating every time when a cryptographic operation is required for certificates containing *Entrust* certificate OID details, and remove the default registration key value.	**Value Name:** NonRepudiationOID **Value:** Empty **Default:** No override

Description	Value
Ignore Silent Mode Determines if the *Token Logon* window is displayed even when the application calls the CSP/KSP in silent mode.	**Value Name:** IgnoreSilentMode **Values:** > 1 (True) - Display the Token Logon window even in silent mode > 0 (False) - Respect silent mode **NOTE** Set to True when the SafeNet RSA KSP must use SHA-2 to enroll a CA private key to a token. **Default:** 0 (False)
PIN Pad Notify Determines if the Pin Pad notification is displayed as balloon or in a window.	**Value Name:** PinPadNotify **Values:** > 0 - Show window > 1 - Show balloon > 2 - No notification **Default:** 0 (Show window)
No Pin Pad Determines whether or not the PIN Pad reader is used as a regular smart card reader. SAC UI requires to enter user credentials.	**Value Name:** NoPinPad **Values:** > 0 - Disabled > 1 - Enabled **Default:** 0 (Disabled)
Force Create Without Touch Sens Determines whether to ignore Touch Sense configuration of the device when creating keys. This setting is applicable only to newly created keys.	**Value Name:** ForceCreateWithoutTouchSens **Values:** > 0 - Touch Sense configuration of the device applies when creating the keys. > 1 - Touch Sense configuration of the device is ignored for the newly created keys. New keys are created as standard keys with Touch Sense disabled for them. **Default:** 0

Description	Value
Allow Sign Final Pin Check Displays SAC digital signature pin pop up in case of multi part signing with sign only key on a CC card. **NOTE** To display digital signature pin pop up, EnablePrompt must be set to 1 in [GENERAL] section.	**Value Name:** AllowSignFinalPinCheck **Values:** > 0- SAC does not display digital signature pin pop up in case of multi part signing. Reuse Current Token Name > 1- SAC displays the digital signature pin pop up in case of multi part signing. **Default:** 0

Token-Domain Password Settings

The following settings are written to the **SyncPin** section in the file `/etc/eToken.conf`.

Description	Value
Synchronize with Domain Password Determines if synchronization is enabled between the eToken password and the domain password.	**Value Name:** Domain **Values:** > Name of the domain (written without a suffix) whose password is synchronized with the Token Password > None - Password synchronization is not enabled **Default:** None

Initialization Settings

The following settings are written to the **INIT** section in the file `/etc/eToken.conf`.

> **NOTE**   All setting in this section are not relevant to IDPrime cards, except for the *LinkMode* setting.

Description	Value
Maximum Token Password Retries Defines the default number of consecutive failed logon attempts that lock the token.	**Value Name:** UserMaxRetry **Values:** 1-15 **Default:** 15
Maximum Administrator Password Retries Defines the default number of consecutive failed administrator logon attempts that lock the token.	**Value Name:** AdminMaxRetry **Values:** 1-15 **Default:** 15
Legacy Format Version Defines the default token format.	**Value Name:** Legacy-Format-Version **Values:** > 0 - Tokens are formatted as backwardly compatible to eToken PKI Client 3.65 (CardOS tokens only) > 4 - Tokens are not formatted as backwardly compatible, and password quality settings can be saved on the token (CardOS tokens only) > 5 - Format includes new RSA behavior that is not controlled by key size; each key is created in a separate directory (CardOS 4.20B FIPS or Java Card-based tokens only) **Default:** > 4, for CardOS tokens > 5, for 4.20B FIPS and Java Card -based tokens
Default Token Name Defines the default Token Name written to tokens during initialization.	**Value Name:** DefaultLabel **Value:** String **Default:** My Token
API: Keep Token Settings When initializing the token using SDK, determines if the token is automatically re-initialized with its current settings. **NOTE** If selected, this setting overrides all other initialization settings.	**Value Name:** KeepTokenInit **Values:** > 1 (True) - Use current token settings > 0 (False) - Override current token settings **Default:** 0 (False)

Description	Value
Automatic Certification When initializing the token using the SDK. If the token has FIPS or Common Criteria certification, the token is automatically initialized with the original certification.	**Value Name:** Certification **Values:** > 1(True) - initialize the token with the original certification > 0 (False) - initialize the token without the certification **Default:** 1 (True) **NOTE** - Previous to SAC 8.2, the default setting value is 0 (False). As CardOS 4.2 does not support both FIPS and RSA-2048, failure to take this into account may lead to token initialization failure when using PKCS#11. - To prevent this, ensure that the default is set to False, or else ensure that the application provides both the required FIPS and RSA-2048 settings.
API: Private Data Caching If using an independent API for initialization, and if *Enable Private Cache* is selected, determines the token's private data cache default behavior.	**Value Name:** PrvCachingMode **Values:** > 0 - Always > 1 - While user is logged on > 2 - Never **Default:** 0 (Always)
Enable Private Data Caching Modification Determines if the token's Private Data Caching mode is can be modified after initialization.	**Value Name:** PrvCachingModify **Values:** > 1 (True) - Can be modified > 0 (False) - Cannot be modified **Default:** 0 (False)

Description	Value
Private Data Caching Mode If *Enable Private Data Caching Modification* is selected, determines who has rights to modify the token's Private Data Caching mode.	**Value Name:** PrvCachingOwner **Values:** > 0 - Admin > 1 - User **Default:** 0 (Admin)
API: RSA Secondary Authentication Mode If using an independent API for initialization, determines the default behavior for protecting RSA private keys on the token.	**Value Name:** 2ndAuthMode **Values:** > 0 - Never > 1 - Prompt on application request > 2 - Always prompt user > 3- Always > 4 - Token authentication on application request **Default:** 0 -(Never)
Enable RSA Secondary Authentication Modified Determines if the token's RSA secondary authentication can be modified after initialization.	**Value Name:** 2ndAuthModify **Values:** > 1 (True) - Can modify > 0 (False) - Cannot modify **Default:** 0 (False)
Use the same token and administrator passwords for digital signature operations.	**Value Name:** LinkMode **Values:** > 1 (True) - Linked > 0 (False) - Unlinked **Default:** 0 (False)

SafeNet Authentication Client Tools UI Initialization Settings

The following settings are written to the **AccessControl** section in the file `/etc/eToken.conf`.

Description	Value
Enable Advanced View Button Determines if the *Advanced View* icon is enabled in SAC Tools	**Value Name:** AdvancedView **Values:** **>** 1 - Selected **>** 0 - Not selected **Default:** 1

The following settings are written to the **InitApp** section in the file `/etc/eToken.conf`.

Description	Value
Default Token Password Defines the default Token Password.	**Value Name:** DefaultUserPassword **Values:** String **Default:** 1234567890
Enable Change Password on First Logon Determines if the *Token Password must be changed on first logon* option can be changed by the user in the T*oken Initialization* window. NOTE This option is selected by default. If the option is de-selected, it can be selected again.	**Value Name:** MustChangePasswordEnabled **Values:** **>** 1 - Selected **>** 0 - Not selected **Default:** 1
Change Password on First Logon Determines if the *Token Password must be changed on first logon* option is selected by default in the *Token Initialization* window.	**Value Name:** MustChangePassword **Value:** **>** 1 - Selected **>** 0 - Not selected **Default:** 1

Description	Value
Private Data Caching If *Enable Private Cache* is selected, determines the token's private data cache default behavior. **NOTE** Can be set in SAC Tools. This option is not supported by IDPrime cards.	**Value Name:** PrivateDataCaching **Values:** > 0 - (fastest) private data is cached when used by an application while the user is logged on to the token, and erased only when the token is disconnected > 1 - private data is cached when used by an application while the user is logged on to the token, and erased when the user logs off or the token is disconnected > 2 - private data is not cached **Default:** 0
RSA Secondary Authentication Mode Defines the default behavior for protecting RSA private keys on the token **NOTE** Can be set in SAC Tools. This option is not supported by IDPrime cards.	**Value Name:** RSASecondaryAuthenticationMode **Values:** > 0 - Never > 1 - Prompt user on application request > 2 - Always prompt user > 3 - Always > 4 - Token authentication on application request **Default:** 0
Reuse Current Token Name Determines if the token's current Token Name is displayed as the default Token Name when the token is re initialized.	**Value Name:** ReadLabelFromToken **Values:** > 1 -The current Token Name is displayed > 0 -The current Token Name is ignored **Default:** 1
Hide Create Administrator Password Fields Determines if Create Administrator Password fields are hidden/ visible in the Password Settings window.	**Value Name:** HideInitCreateAdmin **Values:** > 1: Create Administrator Password fields are hidden > 0: Create Administrator Password fields are visible. **Default:** 0

SafeNet Authentication Client Tools UI Settings

The following settings are written to the **UI** section in the file `/etc/eToken.conf`.

Use Default Password Determines if the *Change Password on First Logon* process assumes the current Token Password is the default (defined in the Default Token Password), and does not prompt the user to supply it.	**Value Name:** UseDefaultPassword **Values:** > 1 (True) - The default Token Password is automatically entered in the password field > 0 (False) -The default Token Password is not automatically entered in the password field **Default:** 0 (False)
Password Term Defines the term used for the token's user password. **NOTE** If a language other than English is used, ensure that the Password Terms are translated.	**Value Name:** PasswordTerm **Values** (String): > Password > PIN > Passcode > Passphrase **Default:** Password
Decimal Serial Number Determines if the Token Information window displays the token serial number in hexadecimal or in decimal format.	**Value Name:** ShowDecimalSerial **Values:** > 1 (True) -Displays the serial number in decimal format > 0 (False) -Displays the serial number in hexadecimal format **Default:** 0

Enable Tray Icon Determines if the application tray icon is displayed when SAC is started.	**Value Name:** ShowInTray **Values:** > 0 - Never Show > 1 - Always Show **Default:** Always show
Enable Connection Notification Determines if a notification balloon is displayed when a token is connected or disconnected.	**Value Name:** ShowBalloonEvents **Values:** > 0 - Not Displayed > 1 - Displayed **Default:** 0
Enable Logging Control Determines if the *Enable Logging /Disable Logging* button is enabled in the *Client Settings > Advanced* tab.	**Value Name:** AllowLogsControl **Values:** > 1 -Enabled > 0 -Disabled **Default:** 1
Home URL Overwrites the SafeNet home URL in SAC Tools.	**Value Name:** HomeUrl **Values** (String): Valid URL **Default:** Thales' (CPL) home URL
Enable Certificate Expiration Warning Determines if a warning message is displayed when certificates on the token are about to expire.	**Value Name:** CertificateExpiryAlert **Values:** > 1 (True) - Notify the user > 0 (False) - Do not notify the user **Default:** 1 (True)

Ignore Expired Certificates Determines if expired certificates are ignored, and no warning message is displayed for expired certificates	**Value Name:** IgnoreExpiredCertificates **Values:** > 1 - Expired certificates are ignored > 0 - A warning message is displayed if the token contains expired certificates **Default:** 0
Certificate Expiration Verification Frequency Defines the minimum interval, in days, between certificate expiration date verifications.	**Value Name:** UpdateAlertMinInterval **Values:** > 0 **Default:** 14 days
Certificate Expiration Warning Period Defines the number of days before a certificate's expiration date during which a warning message is displayed.	**Value Name:** ExpiryAlertPeriodStart **Values:** > =0 (0 = No warning) **Default:** 30 days
Warning Message Title Defines the title to display in certificate expiration warning messages.	**Value Name:** AlertTitle **Values:** String **Default:** SafeNet Authentication Client
Certificate Will Expire Warning Message Defines the warning message to display in a balloon during a *Certificate Expiration Warning Period*.	**Value Name:** FutureAlertMessage **Values:** String **Default:** A certificate on your token expires in $EXPIRE_IN_DAYS days.
Certificate Expired Warning Message Defines the warning message to display in a balloon if a certificate's expiration date has passed.	**Value Name:** PastAlertMessage **Values:** String **Default:** Update your token now.

Warning Message Click Action Defines what happens when the user clicks the message balloon.	**Value Name:** AlertMessageClickAction **Values:** > 0 - No action > 1 - Show detailed message > 2 - Open website **Default:** 0
Detailed Message If *Show detailed message* is selected in *Warning Message Click Action* setting, defines the detailed message to display.	**Value Name:** ActionDetailedMessage **Values:** String **Default:** NA
Website URL If *Open website* is selected in the *Warning Message > Click* setting, defines the URL to display.	**Value Name:** ActionWebSiteURL **Values** (string): Website address **Default:** NA
Enable Password Expiration Notification Determines if a pop-up message is displayed in the system when the Token Password is about to expire.	**Value Name:** NotifyPasswordExpiration **Values:** > 1 (True)- A message is displayed > 0 (False) - A message is not displayed **Default:** 1 (True)
Display Virtual Keyboard Determines if SafeNet's keystroke-secure Virtual Keyboard replaces standard keyboard entry of password fields in the following windows: > Token Logon > Change Password **NOTE** The virtual keyboard supports English characters only.	**Value Name:** VirtualKeyboardOn **Values:** > 1 (True)- Virtual keyboard on > 0 (False) - Virtual keyboard off **Default:** 0 (False)

Password Policy Instructions If not empty, defines a string that replaces the default password policy description displayed in the *Unlock and Change Password* windows.	**Value Name:** PasswordPolicyInstructions **Values:** String
Define Initialization Mode Select this option if you want the *Initialization Options* window (first window displayed when initializing a device) to be ignored.	**Value Name:** DefInitMode **Values:** > 0 - Display the 'Initialization Options' window > 1 - Set Preserve Mode > 2 - Set Configure Mode **Default:** 0
Import Certificate Chain Determines if the certificate chain is imported to the token.	**Value Name:** ImportCertChain **Values:** > 0 - Do not import certificate chain > 1 - Import certificate chain > 2- User selects import behavior **Default:** 0

Token Password Quality Settings

The following settings are written to the **PQ** section in the file `/etc/eToken.conf.`

> **NOTE**   These settings are not relevant to IDPrime cards and eToken 5110 CC, as the *Password Quality* settings reside on the card itself.

Description	Value
Password - Include Non ASCII Characters Determines if the password can be included for non-ASCII characters. **NOTE** Applicable for IDPrime cards only.	**Value Name**: pqNonAscii **Values**: > 0: Permitted > 1: Forbidden > 2: Mandatory **Default**: 0
Password - Number Of Different Repeating Characters Determines the number of different characters that can be repeated at least once.	**Value Name**: pqNumDiffCharRepeat **Values**: >= 0 (0 = No check) **Default**: 0
Password - Maximum Number A Character Can Appear Determines the maximum number a character can appear.	**Value Name**: pqMaxNumCharAppear **Values**: >= 0 (0 = No check) **Default**: 0
Password - Maximum Number Of Characters In A Sequence Determines the maximum number of characters in a sequence. For example: If the value is set to 4, the sequence 1,2,3,4,a,5 is allowed but 1,2,3,4,5,a is not allowed.	**Value Name**: pqMaxNumCharSequence **Values**: >= 0 (0 = No check) **Default**: 0
Password - Maximum Adjacent Repetitions Of A Character Determines the maximum number a character can be repeated in adjacent positions. **NOTE** If pqMaxNumCharRepeatPos = 0, then the value of pqMaxRepeated is applicable.	**Value Name**: pqMaxNumCharRepeatPos **Values**: >= 0 (0 = No check) **Default**: 0
Password - Minimum Length Defines the minimum password length. **NOTE** Can be set in SAC Tools.	**Value Name:** pqMinLen **Values:** >=4 **Default:** 6

Description	Value
Password - Maximum Length Defines the maximum password length. > **NOTE** Can be set in SAC Tools.	**Value Name:** pqMaxLen **Values:** Cannot be less than the Password Minimum Length **Default:** 16
Password - Maximum Usage Period Defines the maximum number of days a password is valid. > **NOTE** Can be set in SAC Tools. > **NOTE** This parameter is *Day Sensitive* i.e. the system counts the day's and not the hour in which the user made the change.	**Value Name:** pqMaxAge **Values:** >=0 (0 =No expiration) **Default:** 0
Password - Minimum Usage Period Defines the minimum number of days between password changes. > **NOTE** Can be set in SAC Tools.	**Value Name:** pqMinAge **Values:** >=0 (0 = No minimum) **Default:** 0
Password - Expiration Warning Period Defines the number of days before expiration during which a warning is displayed. > **NOTE** Can be set in SAC Tools.	**Value Name:** pqWarnPeriod **Values:** >=0 (0 = No warning) **Default:** 0
Password - History Size Defines the number of recent passwords that must not be repeated. > **NOTE** Can be set in SAC Tools.	**Value Name:** pqHistorySize **Values:** >= 0 (0 = No minimum) **Default:** 10

Description	Value
Password - Maximum Consecutive Repetitions Defines the maximum number of consecutive times a character can be used in a password. > **NOTE** Can be set in SAC Tools.	**Value Name:** pqMaxRepeated **Values:** 0 - 16 (0 = No maximum) **Default:** 3
Password - Complexity Determines if there is a minimum number of character types that must be included in a new Token Password. The character types are upper-case letters, lower-case letters, numerals, and special characters. > **NOTE** Can be set in SAC Tools.	**Value Name:** pqMixChars **Values:** > 1 - A minimum of 2 or 3 types must be included, as defined in the Password-Minimum Mixed Character Types setting > 0 -The rule for each character type is defined in the character type's Include setting **Default:** 1
Password - Minimum Mixed Character Types Defines the minimum number of character types that must be included in a new Token Password. The character types are upper-case letters, lower-case letters, numerals, and special characters. > **NOTE** > - Applies only when the *Password - Complexity* setting is set to *Standard complexity*. > - Can be set in SAC Tools.	**Value Name:** pqMixLevel **Values:** > 0 - At least 3 character types > 1 - At least 2 character types **Default:** 0
Password - Include Numerals Determines if the password can include numerals. > **NOTE** > - Applies only when the *Password - Complexity* setting is set to *Manual complexity*. > - Can be set in SAC Tools.	**Value Name:** pqNumbers **Values:** > 0 -Permitted > 1 - Forbidden > 2 - Mandatory **Default:** 0

Description	Value
Password - Include Upper-Case Determines if the password can include upper-case letters. **NOTE** - Applies only when the *Password - Complexity* setting is set to *Manual complexity*. - Can be set in SAC Tools.	**Value Name:** pqUpperCase **Values:** > 0 - Permitted > 1 - Forbidden > 2 - Mandatory **Default:** 0
Password - Include Lower-Case Determines if the password can include lower-case letters. **NOTE** - Applies only when the *Password - Complexity* setting is set to *Manual complexity*. - Can be set in SAC Tools.	**Value Name:** pqLowerCase **Values:** > 0 - Permitted > 1 - Forbidden > 2 - Mandatory **Default:** 0
Password - Include Special Characters Determines if the password can include special characters, such as @,!, &. **NOTE** - Applies only when the *Password - Complexity* setting is set to *Manual complexity*. - Can be set in SAC Tools.	**Value Name:** pqSpecial **Values:** > 0 - Permitted > 1 - Forbidden > 2 - Mandatory **Default:** 0
Password Quality Check on Initialization Determines if the *Password Quality* settings are checked and enforced when a token is initialized **NOTE** It is recommended that this policy must not be set when tokens are enrolled using SafeNet Authentication Manager.	**Value Name:** pqCheckInit **Values:** > 1 (True) -The password quality is enforced > 0 (False) - The password quality is not enforced **Default:** 0

Description	Value
Password Quality Owner Defines the owner of the *Password Quality* settings on a re initialized token, and defines the default of the *Password Quality Modifiable* setting.	**Value Name:** pqOwner **Values:** > 0 - Administrator > 1 - User **Default:** > 0 - for tokens with an Administrator Password > 1 - for tokens without an Administrator Password
Enable Password Quality Modification Determines if the *Password Quality* settings on a newly initialized token can be modified by the owner.	**Value Name:** pqModifiable **Values:** > 1 (True) - The password quality can be modified by the owner > 0 (False) - The password quality cannot be modified by the owner **Default:** > 1 (True) - for administrator owned tokens > 0 (False) - for user owned tokens

SafeNet Authentication Client Tools UI Access Control List

Access Control Properties determine which features are enabled in the SafeNet Authentication Client Tools and Tray Menu.

The following settings are written to the **AccessControl** section in the file `/etc/eToken.conf`.

Access Control Feature	Value
All access control features are listed in below table.	**Values:** > 1 (True) - The feature is enabled. > 0 (False) - The feature is disabled. **Default:** 1(True), except where indicated in the table

> **NOTE**   All access control features are enabled by default, except where indicated in the table.

Access Control Feature	Value Name	Description
Change Digital Signature PUK	ChangeDigitalSignaturePUK	Enables/Disables the Change Digital Signature PUK feature in SafeNet Authentication Client Tools.
Change Digital Signature PIN	ChangeDigitalSignaturePIN	Enables/Disables the Change Digital Signature PIN feature in SafeNet Authentication Client Tools.
Set Digital Signature PIN	SetDigitalSignaturePIN	Enables/Disables the Set Digital Signature PIN feature in SafeNet Authentication Client Tools.
Crypto Notification Timeout	CryptoNotificationTimeout	Enables/Disables the notification: "The process may take a while...." **NOTE** By default, this feature is disabled.
Rename Token	RenameToken	Enables/Disables the *Rename Token* feature in SafeNet Authentication Client Tools.
Change Token Password	ChangePassword	Enables/Disables the *Change Token Password* feature in SafeNet Authentication Client Tools.

Access Control Feature	Value Name	Description
Unlock Token	UnlockEtoken	Enables/Disables the *Unlock Token* feature in SafeNet Authentication Client Tools.
Delete Token Content	ClearEtoken	Enables/Disables the *Delete Token Content* feature in SafeNet Authentication Client Tools.
View Token Information	ViewTokenInfo	Enables/Disables the *View Token Information* feature in SafeNet Authentication Client Tools.
Help	ShowHelp	Determines if the user can open the *Help file* in SafeNet Authentication Client Tools.
Advanced View	OpenAdvancedView	Determines if the user can open the *Advanced View* in SafeNet Authentication Client Tools.
Reader Settings	ManageReaders	Enables/Disables the *Reader Settings* feature in SafeNet Authentication Client Tools.
Initialize Token	InitializeEtoken	Enables/Disables the *Initialize Token* feature in SafeNet Authentication Client Tools.

Access Control Feature	Value Name	Description
Import Certificate	ImportCertificate	Enables/Disables the *Import Certificate* feature in SafeNet Authentication Client Tools.
Reset Default Certificate Selection	ClearDefaultCert	Enables/Disables the *Reset Default Certificate Selection* feature in SafeNet Authentication Client Tools.
Delete Certificate	DeleteCertificate	Enables/Disables the *Delete Certificate* feature in SafeNet Authentication ClientTools.
Export Certificate	ExportCertificate	Enables/Disables the *Export Certificate* feature in SafeNet Authentication Client Tools.
Copy Certificate Data to Clipboard	CopyCertificateData	Enables/Disables the *Copy Certificate Data to Clipboard* feature in SafeNet Authentication Client Tools.
Set Certificate as Default	SetCertificateAsDefault	Enables/Disables the *Set Certificate as Default* feature in SafeNet Authentication Client Tools.

Access Control Feature	Value Name	Description
Set Certificate as Auxiliary	SetCertificateAsAuxilary	Enables/Disables the *Set Certificate as Auxiliary* feature in SafeNet Authentication Client Tools.
Log On as Administrator	LoginAsAdministrator	Enables/Disables the *Log On as Administrator* feature in SafeNet Authentication Client Tools.
Change Administrator Password	ChangeAdministratorPassword	Enables/Disables the *Change Administrator Password* feature in SafeNet Authentication Client Tools.
Set Token Password	SetUserPassword	Enables/Disables the *Set Token Password* feature in SafeNet Authentication Client Tools.
Token Password Retries	AllowChangeUserMaxRetry	Enables/Disables the *Logon retries before token is locked* feature (for the Token Password) in SafeNet Authentication Client Tools.
Administrator Password Retries	AllowChangeAdminMaxRetry	Enables/Disables the *Logon retries before token is locked* feature (for the Administrator Password) in SafeNet Authentication Client Tools.

Access Control Feature	Value Name	Description
Advanced Initialization Settings	OpenAdvancedModeOfInitialize	Enables/Disables the *Advanced* button in the Token Initialization window in SafeNet Authentication Client Tools.
Change Initialization Key during Initialization	ChangeInitializationKeyDuringInitialize	Enables/Disables the *Change Initialization key* button in the Advanced Token Initialization Settings window in SafeNet Authentication Client Tools
Common Criteria Settings	CommonCriteriaPassword Setting	Enables/Disables the *Common Criteria* option in the Certification combo box.
System Tray - Unlock Token	TrayIconUnlockEtoken	Enables/Disables the *Unlock Token* feature in the SafeNet Authentication Client Tray Menu.
System Tray - Delete Token Content	TrayIconClearEtoken	Enables/Disables the *Delete Token Content* feature in the SafeNet Authentication Client Tray Menu. **NOTE** By default, this feature is Disabled.

Access Control Feature	Value Name	Description
System Tray - Change Token Password	TrayIconChangePassword	Enables/Disables the *Change Token Password* feature in the SafeNet Authentication Client Tray Menu.
System Tray - Synchronize Domain-Token Passwords	SyncDomainAndTokenPass	Enables/Disables the *Synchronize Domain Token Passwords* feature in the SafeNet Authentication Client Tray Menu.
System Tray - Tools	OpenEtokenProperties	Enables/Disables the *Tools* menu item (open SafeNet Authentication Client Tools) in the SafeNet Authentication Client Tray Menu.
System Tray - About	About	Enables/Disables the *About* menu item in the SafeNet Authentication Client Tray Menu.
Enable Change IdenTrust Identity	IdenTrustChangePassword	Enables/Disables the *Change IdenTrust PIN* feature in SafeNet Authentication Client Tools.
Enable Unblock IdenTrust Passcode	IdenTrustUnlock	Enables/Disables the *Unlock IdenTrust* feature in SafeNet Authentication Client Tools.

Access Control Feature	Value Name	Description
Delete Data Object	DeleteDataObject	Enables/Disables the *Delete Data Object* feature in SafeNet Authentication Client Tools.
Allow One Factor	AllowOneFactor	Enables/Disables the *Allow One Factor* feature in the Advanced Token Initialization Settings window in SafeNet Authentication Client Tools.

Security Settings

The following settings are written to the **Crypto** section in the file `/etc/eToken.conf`.

Description	Value

Key Management
Defines key creation, export, unwrap, and off-board crypto policies.
 	
> **NOTE**
> Edit the `/etc/eToken.conf` file as follows to migrate classic client customers to SAC:
> `[GENERAL]`
> `[Crypto]`
> `Key-Management-`
> `Security=Compatible`
> This ensures that any deprecated keys sizes or algorithms will remain supported on SAC.
 	
For more information, refer to "SafeNet Authentication Client Security Enhancements" on page 70.
	Value Name:
	Key-Management-Security
	
	Values: (String)
	> Compatible - has no effect, current behavior is kept
	> Optimized:
	  • Disable the generation or creation of exportable keys.
	  • Disable the exporting of keys, regardless of how they were generated.
	  • Disable the unwrap-PKCS1.5 and unwrap-AES-CBC on hardware tokens (session enable).
	> Strict:
	  • Disable the generation or creation of exportable keys.
	  • Disable the exporting of keys, regardless of how they were generated.
	  • Disable the unwrap-PKCS1.5 and unwrap-AES-CBC operations.
	  • Disable any usage of symmetric keys off-board including unwrap.
	
	Default: Optimized

Description	Value

Disable-Crypto
Unsupported Cryptographic Algorithms and Features.
 	
NOTE
Edit the `/etc/eToken.conf` file as follows to migrate classic client customers to SAC:
`[GENERAL]`
`[Crypto]`
`Disable-Crypto=none`
This ensures that any deprecated keys sizes or algorithms will remain supported on SAC.
 	
For more information, refer to "SafeNet Authentication Client Security Enhancements" on page 70.
	Value Name: Disable-Crypto
	
	Values: (String)
	> None - All SAC cryptographic algorithms and features are supported.
	  • This is the default value for SAC Mac versions below 10.2.
	  • Setting this value will allow SAC to be compatible with SAC Mac 10.2 and below.
	  • It is strongly recommended to read "Security Recommendations" on page 70 before applying legacy values.
	> Obsolete - A list of restricted and deprecated cryptographic algorithms and features.
	  The following are deprecated: MD5, RC2, RC4, DES, 2DES, GenericSecret<112, RSA-RAW, RSA<2048, ECC<224, ECB, Sign-SHA1.
	> Manual - Create your own list of algorithms. (Refer to the description below).
	
	Default: Obsolete

The following can be disabled:

> **Algorithms:** RSA, ECC, DES, 2DES, 3DES, AES, RC2, RC4, GenericSecret

> **Hash types:** MD5, SHA1, SHA2

> **Padding types:** RAW, PKCS1, OAEP, PSS

> **Cipher modes:** ECB, CBC, CTR, CCM

> **Mechanisms:** MAC, HMAC, ECDSA, ECDH

> **Operations:** Encrypt, Decrypt, Sign, Verify, Generate, Derive, Wrap, Unwrap, Digest, Create (keys only)

> **Weak key size:** RSA<2048

> **Object types:**

  • HWEF – elementary file (EF) objects (used by eToken devices for storing exportable symmetric keys and symmetric keys without on-board implementation)

  • HWALL – all types of objects implemented on token (Base Security Object (BSO) and EF),

Example of a manual configuration: Encrypt-DES-ECB, Sign-3DES-MAC, DES-CTR, HMAC-MD5, HMAC-SHA1, HMAC-SHA2, DES-CBC, Unwrap-DES-ECB, RSA-PKCS1-MD5, Verify-RSAPSS-SHA2, AES-CTR, AES-MAC, Decrypt-RC2, Wrap-ECB.

Log Settings

The following settings are written to the **Log** section in the file `/etc/eToken.conf`.

Description	Value
Enabled Determines if the *SafeNet Authentication Client Log* feature is enabled.	**Value Name:** Enabled **Value:** > 1 - Enabled > 0 - Disabled **Default:** 0 (Disabled)
Days Defines the number of days log files will be saved from the time the log feature was enabled.	**Value Name:** Days **Value:** Enter the number of days (numerical). **Default:** 1 day
MaxFileSize Defines the maximum size of an individual log file. Once the maximum file size is reached, SAC removes older log records to allow saving newer log information.	**Value Name:** MaxFileSize **Value:** Enter a value in Bytes. **Default:** 2000000 (Bytes) (Approximately 2MB)
TotalMaxSizeMB Defines the total size of all the log files when in debug mode (Megabytes).	**Value Name:** TotalMaxSizeMB **Value:** Enter a value in Megabytes. **Default:** 0 (Unlimited)

Description	Value
ManageTimeInterval Defines how often the *TotalMaxSize* parameter is checked to ensure that the total maximum is not exceeded.	**Value Name:** ManageTimeInterval **Value:** Enter a value in minutes (numerical). **Default:** 60 minutes

CHAPTER 7:   Security Recommendations

The information provided in this chapter helps you maintain a secured SAC environment and keep your information safe.

SafeNet Authentication Client Security Enhancements

User/Administrator smart card are password protected. To avoid password leakage, Thales recommends the following:

> Use PIN Pad readers - user passwords do not pass through a computers memory when using a PIN Pad reader.

> Use devices configured to support secured messaging - secured messaging protects the transfer of data between the middleware and the device.

> Protect the device from unauthorized usage.

> Ensure the device is disconnected when not in use.

> Configure restrictive password policies. For more information, refer to the *SafeNet Authentication Client Mac User Guide*.

- Change the default administrator password.

- For devices running the eToken applet, change the default Initialization Key (this protects devices from unwanted initialization).

- If the device was enrolled by an administrator (on behalf of a user), use the 'Token password must be changed on first logon' option.

- For supported devices use the on-board password quality settings (use the 'Enforce password quality settings' option).

> The recommended password strength is:

- User PIN should include at least 8 characters of different types.

- Admin PIN should include at least 16 characters of different character types.

- The Friendly Admin Password should include at least 16 characters of different types. (For more details on the Friendly Admin Password, refer to *SafeNet Authentication Client Mac User Guide*).

- Digital Signature PUK, when using a friendly name, this should include at least 16 characters of different types.

- For devices running the IDPrime applet, the 3DES random key may be used instead of the administrator password.

- Use the password validity period combined with password history options.

For more information on how to configure password policy settings, refer to "Token Password Quality Settings" on page 53, as well as the Token Initialization chapter of the *SafeNet Authentication Client Mac User Guide* and the Password Recommendations section of the *SafeNet Authentication Client Mac Release Notes*.

> **NOTE**   Character types include upper case, lower case, numbers, and special characters.

Enforcing Restrictive Cryptographic Policies

To allow organizations to enforce restrictive cryptographic policies when using SafeNet smart card and USB tokens, the following enhancements are introduced:

> Key Management Security Policy

> Disable Cryptographic Algorithm Policy

For more details, refer to "Security Settings" on page 65.

The motivation behind these enhancements:

> Legacy cryptographic schemes cause organizations to fail current compliance requirements or expose cryptographic weakness associated with obsolete algorithms and mechanisms.

  The following enhancements made to SafeNet Authentication Client to allow organizations to block the use of such schemes, according to organizational policies.

  • Enabling symmetric keys wrapping with other symmetric keys using GCM and CCM modes of operation.

  • Preventing legacy algorithms from being used by adding a key wrapping policy that enforces the usage of only GCM and CCM modes of operation for symmetric encryption, and PKCS#1 v2.1 padding for RSA encryption.

> SafeNet introduced a new mechanism that allows administrators to prevent the use of legacy or obsolete algorithms by third-party applications. These cryptographic algorithms conform to the National Institute of Standards and Technology (NIST), preventing third-party applications from using legacy or obsolete algorithms.

> **NOTE**   Once a restrictive policy is set, the use of SafeNet Authentication Client with the above algorithms is blocked.
> - This may have implications on the way in which the third-party's applications currently work.
> - Administrators must make sure that the third-party applications used by the organization are configured accordingly, and do no use any of the algorithms listed above, as they will be blocked.

Creating Symmetric Key Objects using PKCS#11

The following are performed as part of SafeNet Authentication Client security enhancement campaign:

1. Protected memory is used when working with the private cache between PKCS#11 API calls. Private cache is unlocked to retrieve data and then locked immediately after retrieving the data to ensure that there is no sensitive data in the private cache. This ensures that the key cannot be revealed in plain text.

2. Sensitive data is securely zeroed prior to freeing up the memory.

3. AES and Generic symmetric key files are created with Secured Messaging (SM) protection, so that the smart card transport layer does not contain any APDU data with plain symmetric key material.

For SM to support the AES/3DES and Generic symmetric keys in SAC 10.8, the keys must be created on an eToken Java device that is initialized in FIPS/CC mode. Applying SM to symmetric keys changes the object format on the smart card, resulting in the keys not being backward compatible.

> **NOTE**   Keys that are created with previous SAC versions or on eToken Java devices which are formatted in non-FIPS/CC mode are not protected by SM.
> AES/3DES keys that are created using the `CKA_SENSITIVE = TRUE` and `CKA_EXTRACTABLE = FALSE` attributes are backward compatible (BS Object keys).

Ensuring a Secured SAC Environment

This section provides short guidelines on how to maintain a safe Mac computer environment. The information is based on the security recommendations defined by Apple.

Software Updates

The best way to keep your Mac secure is to run the latest software. When new updates are available, macOS sends you a notification. Just accept the updates with a click and they download automatically. macOS checks for new updates every day, so it's easy to always have the latest and safest version.

System Security Control

System Preferences contains privacy controls for location sharing and diagnostic information sharing. Safari preferences include a privacy window that allows you to limit or block cookies and limit website access to location services.

Malware Awareness

Innocent-looking files downloaded over the Internet may contain dangerous malware in disguise. That's why files downloaded using Safari, Mail, and Messages are screened to determine if they contain applications. If they do, Mac OS send an alert and warns you the first time you open one. It is up to your discretion to open or cancel the application. And, if a file contains software identified as malicious, Mac OS offers to move it to the trash.