SafeNet Authentication Client (Mac)

Version 10.1 (GA)

Administrator Guide

gemalto
security to be free

Product Version: 10.1 Mac (GA)

Document Number: 007-013726-002, Rev. A

Release Date: November 2017

Support Contacts

We work closely with our reseller partners to offer the best worldwide technical support services. Your reseller is the first line of support when you have questions about products and services. However, if you require additional assistance you can contact us directly at:

If you encounter a problem while installing, registering or operating this product, please make sure that you have read the documentation. If you cannot resolve the issue, contact your supplier or Gemalto Customer Support. Gemalto 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 Gemalto and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Contact Method	Contact Information
Customer Support Portal	https://supportportal.gemalto.com Existing customers with a Technical Support Customer Portal account can log in to manage incidents, get the latest software upgrades, and access the Gemalto Knowledge Base.
Technical Support contact email	technical.support@gemalto.com

Additional Documentation

The following publications are available:

- 007-013725-002 SafeNet Authentication Client 10.1 Mac (GA) User Guide
- 007-013724-002 SafeNet Authentication Client 10.1 Mac (GA) Release Notes (RN)

Table of Contents

Introduction

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

With full backward compatibility and incorporating features from previous middleware versions, SafeNet Authentication Client ensures complete support for all currently deployed eToken and iKey devices, as well as IDPrime MD and .NET smart cards.

Overview

SafeNet Authentication Client is 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.

SafeNet Authentication Client 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 can be created, stored, and used securely from within hardware or software.

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

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

NOTE:
For SafeNet Authentication Client system requirement details and compatibility information, see the SafeNet Authentication Client Mac 10.1 Release Notes.

License Activation

SafeNet Authentication Client 10.1 Mac (GA) is installed by default as non-licensed.

To activate the license perform the following steps:

1. Obtain a valid SAC License Key from SafeNet Customer Service.
2. Activate the license using one of the following procedures:
   - Manual Activation

   See the *Licensing* chapter in the *SafeNet Authentication Client 10.1 Mac (GA) User Guide*.

NOTE:
SafeNet Authentication Client retrieves the license file (SACLicense.lic) automatically, if the license file is located in the following default path:
Mac (per user): /home/<user name>
Mac (per machine): /Users/Shared/SafeNet/SAC

   - SAC Mac Custom Installation - See Chapter 2: Preparing SAC Mac Custom Installation (page 14).

IDPrime MD Applet 4.0

The IDPrime MD Applet 4.0 is Common Criteria certified on IDPrime MD 840 and 3840. These cards can have certain parameters customized in the factory with different values to the standard default profile.

The following parameters can be customized:

- Number and type of key containers
- Support of RSA 4,096-bit key containers (import operation only) - Note: The card needs to be configured by the SAC supported key length.
- Change PIN at first use Secure messaging in contactless mode
- PINs (#1, #3 and #4 only)
- Try Limit
- Unblock PIN (PIN#1 only)
- Policy values
- Properties
- PIN validity period
- Secure messaging in contactless mode

Number and Type of Key Containers

By default, the IDPrime MD 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

API Adjustments

This table below provides a high-level description of the adjustments that can be made to the Standard and Extended PKCS#11 API to work with IDPrime MD CC devices. For more detailed information, see 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 linked mode on the IDPrime MD CC device by using the following registry key: `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\SafeNet\Authentication\SAC\Init - LinkMode (DWORD)` The registry key must be set to 1 and the device must be in the factory initialized state (Admin key = 48 zeros, PUK = 6 zeros) 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 MD 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. See the SafeNet Authentication Client User Guide for details on Friendly Admin Password. The C_SetPIN 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.

Installation

Follow the installation procedures below to install SafeNet Authentication Client 10.1 Mac. Local administrator rights are required to install or uninstall SafeNet Authentication Client.

NOTE:
- If IDGo 800 PKCS#11 is installed, be sure to remove it before installing SAC 10.1 Mac.

Installation Files

The software package provided by SafeNet includes files for installing or upgrading to SafeNet Authentication Client 10.1 Mac (GA). The following installation and documentation files are provided:

File	Description	Use
Mac		
SafeNetAuthenticationClient.10.1.xx.dmg.	Installs SafeNet Authentication Client. The .dmg disk image contains SAC install and uninstall applications.	Installs/Uninstalls SafeNet Authentication Client
SafeNet Authentication Client Customization 10.1.mpkg	This is a separate customer specific installation, used to install the SAC license and configuration file. For details on how to create this file, See "Preparing SAC Mac Custom Installation" on page 14.	This file is created and customized by the administrator, as part of the SAC (Mac) custom license and configuration installation script.
Documentation Files		
007-013724-002_SafeNet Authentication Client_ 10.1_Mac_GA_RN_Revision A	SafeNet Authentication Client 10.1 Customer 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.
007-013725-002_SafeNet Authentication Client_10.1_Mac_GA_User_Guide_Revision A	SafeNet Authentication Client 10.1 Mac (GA) User Guide for Windows	Provides detailed information for the user and system administrator regarding the use of SafeNet Authentication Client for Mac.
007-013726-002_SafeNet Authentication Client_10.1_Mac_GA_Administrator_Guide_Revision A	SafeNet Authentication Client 10.1 Mac (GA) Administrator Guide for Windows (this document)	Provides detailed information for the system administrator regarding the installation, configuration, maintenance, and management of SafeNet Authentication Client for Mac.

Installing SafeNet Authentication Client on Mac OS X

The installation package is SafeNetAuthenticationClient.10.1.x.0.dmg.

To install with the installer:

1. Double click the **SafeNetAuthenticationClient.10.1.x.0.dmg** file.

   A new disk image file is created in the Finder window, including an pkg installation file and an uninstall application.

2. To start the installation, double click **SafeNet Authentication Client 10.1.pkg**.

   The *Welcome to the SafeNet Authentication Client Installer* window opens.

3. Click **Continue**.

The *Software License Agreement* window opens.

4. Click **Continue**.

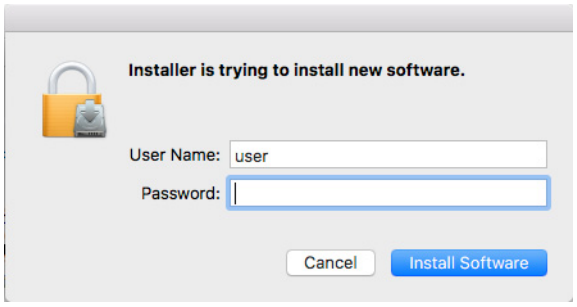
   The *Agreement* window opens.

5. Click **Agree** to accept the software license agreement.

The *Standard Install* window opens.

6. Click **Install**.

   The *Authenticate* window opens.

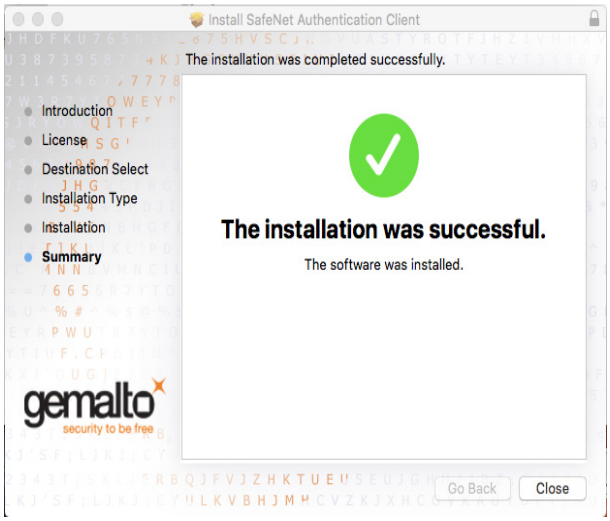


7. Enter Name and Password and click **OK**.

   NOTE:
   Administrator permissions are required to install SafeNet Authentication Client.

SafeNet Authentication Client is installed.

The *Installation completed successfully* screen opens.



8. Click **Close** and then perform a **Restart** (recommended).

   Mac OS X restarts.

9. Log in again to Mac OS X.

Installing SAC from the Mac Terminal

To install from the Mac terminal:

1. Extract the SafeNet Authentication Client 10.1.pkg file from the dmg file.
2. At the location in the terminal in which you extracted the file run sudo installer -pkg ./SafeNet\Authentication\Client\ 10.1.pkg/ -target /
3. Enter your root password when prompted.

   SafeNet Authentication Client 10.1 is installed.

4. It is recommended to Restart Mac OS X.

Preparing SAC Mac Custom Installation

The custom installation script creates an additional customized installation file (SafeNet Authentication Client Customization 10.1.mpkg) with specific license and configuration properties and values.

> **NOTE**:
> Before installing the SafeNet Authentication Client Customization 10.1.mpkg file, you must install SAC 10.1.

To create a custom installation:

1. Copy the **Custom Installation** (packaged with the Mac installation) folder to your Mac PC.
2. Select the **Custom Installation\CustomerConfiguration** folder, and update the contents of the eToken.conf file with the relevant organization properties, and the SacLicense.lic file with the organization's license.
3. From the Mac terminal enter the command:

   ```
   cd [Custom Installation path]\CustomInstallScript.
   ```

4. Run the script:

   ```
   ./createSacCustomInstallation
   ```

   The file **SafeNet Authentication Client Customization 10.1.mpkg** is created in the **Custom Installation\output** folder.
5. The SafeNet Authentication Client Customization 10.1.mpkg file can now be distributed to all users in the organization as an additional SAC 10.1 installation.

Running SafeNet Authentication Client Customization 10.1.mpkg

By running the SafeNet Authentication Client Customization 10.1.mpkg file the following is implemented:

- The **/etc/eToken.conf** file (created by the Mac SAC 10.1 installation) is replaced by the eToken.conf file, located in the **Custom Installation\CustomerConfiguration** folder.
- The **SacLicense.lic** file, located in the **Custom Installation\CustomerConfiguration** folder is copied to the **/Users/Shared/SafeNet/SAC** folder.

Installing the Firefox Security Module on Mac

When SafeNet Authentication Client is installed, it does not install the security module in Firefox. This must be done manually.

To install the security module in Firefox

1. Open **Firefox Preferences > Advanced >Certificates**.
2. On the *Encryption* tab click **Security Devices**.

   The *Device Manager* window opens.
3. Click **Load**.

   The *Load PKCS#11 Device* window opens.
4. In the Module Filename field enter the following string:

   `/usr/local/lib/libeTPkcs11.dylib`

   NOTE:
   - To work with CC devices in unlinked mode, enter the following string for Multi-Slot support: **/usr/local/lib/libidprimepkcs11.0.dylib**
   - For information on how to work with Multi-Slots, see the PKCS#11 Digital Signature PIN Authentication section of the SafeNet Authentication Client User Guide.

   The *Confirm* window opens.
5. Click **OK**.

   The new security module is installed.

Installing the Thunderbird Security Module

When SafeNet Authentication Client is installed, it does not install the security module in Thunderbird. This must be done manually.

To install the security module in Thunderbird

1. Select **Thunderbird > Preferences > Advanced**.
2. On the *Security* tab click **Security Devices**.

   The *Device Manager* window opens.
3. Click **Load**.

   The *Load PKCS#11 Device* window opens.
4. In the Module Filename field enter the following string:

   `/usr/local/lib/libeTPkcs11.dylib`

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

   The *Confirm* window opens.
5. Click **OK**.

   The new security module is installed.

Configuring Acrobat Security Settings

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

To set Adobe Acrobat security settings:

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

   The *Digital ID and Trusted Certificate Settings* window opens.
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 lib/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, see the PKCS#11 Digital Signature PIN Authentication section of the SafeNet Authentication Client User Guide.

To verify the security settings:

1. Select **Tools > Sign & Certify > More Sign & Certify > Manage Trusted Identities**.

   The *Manage Trusted Identities* window opens.
2. Select the contact and click **Details**.

The *Edit Contact* window opens.

3. Select the contact and click **Show Certificate**.

   The *Certificate Viewer* Window opens.

4. Select the **Trust** tab.
   - Trusted settings for the certificate are marked with a green check-mark.
   - Non-trusted settings are marked with a red cross.

Loading the Token PKCS#11 Security Module

To use SafeNet Authentication Client the PKCS#11 security module must be loaded.

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, see the PKCS#11 Digital Signature PIN Authentication section of the SafeNet Authentication Client User Guide.

New Locations of PKCS#11 Security Module and Tokend

El Capitan introduced a new security feature called “System Integrity Protection" (see https://support.apple.com/en-us/HT204899).

This means that the SAC PKCS11 library is installed in a different location for El Capitan (this is not relevant to earlier Mac versions):

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

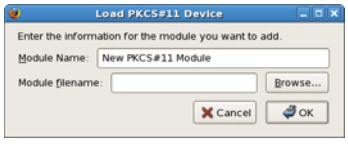
To ensure that the Token PKCS#11 module is loaded:

1. Do one of the following:
   - When working with Firefox, go to **Edit > Preferences > Advanced > Certificates >Security Devices**.
   - When working with Thunderbird, go to **Edit > Preferences > Advanced > Certificates >Security Devices**.

   The *Device Manager* window opens

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

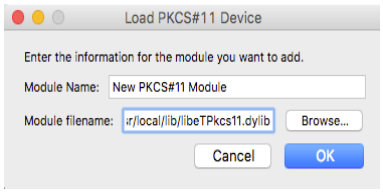
   The *Load PKCS#11 Device* window opens.



3. Do the following:
   - Replace the contents of the Module Name field with eToken.
   - In the Module filename field, enter the following:

     `/usr/lib/libeTPkcs11.dylib`

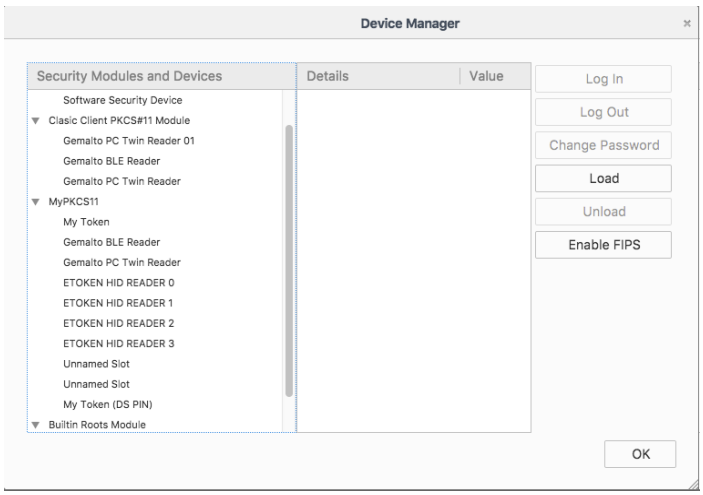
> **NOTE**:
> The Module fields are case sensitive.



4. Click **OK**. The *Confirm* window opens.
5. Click **OK**.

   The Alert window opens.
6. Click **OK**.

   Token is listed in the *Security Modules and Devices* column of the *Device Manager* window.



7. Click **OK** to exit the *Device Manager*.

Upgrading SafeNet Authentication Client on a Mac

It is recommended that earlier versions of SafeNet Authentication Client be upgraded to the latest version on each computer that uses a SafeNet eToken, iKey token, or SafeNet smart card. Local administrator rights are required to upgrade SafeNet Authentication Client.

After upgrading from SAC 9.1 to SAC 10.1 on a Mac, it is recommended that you restart the machine in order for the device to be recognized.

Tokend and Crypto Token Kit Modules

SafeNet Authentication Client 10.1 Mac supports both the Tokend and Crypto Token Kit (CTK) modules that run side-by-side.

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. On macOS 10.12 both architectures are supported, therefore SAC 10.1 Mac supports both architectures by default with the option to disable the Tokend module.

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

Disabling the Tokend Module

To work only with the new CTK, the Tokend module must be disabled by renaming the **/Library/Security/tokend/eToken**d file to: **/Library/Security/tokend/eToken.tokend.disable**

To rename the file using the terminal command: sudo mv **/Library/Security/tokend/eToken.tokend /Library/Security/tokend/eToken.tokend.disable**.

NOTE:
- To re-enable the Tokend module execute the following command: **sudo mv /Library/Security/tokend.disable /Library/Security/tokend**
- Reboot your machine after disabling/renaming the Tokend module.

Disabling the new Crypto Token Kit (CTK) Module

To work only with the Tokend module, the new CTK must be disabled by renaming the **/Library/Frameworks/eToken.framework/Versions/A/SafeNet Authentication Client/Contents/PlugIns** folder.

To rename the folder using the terminal command: **sudo mv /Library/Frameworks/eToken.framework/Versions/A/SafeNet Authentication Client/Contents/PlugIns** to: **/Library/Frameworks/eToken.framework/Versions/A/SafeNet Authentication Client/Contents/PlugIns.disable**

NOTE:
- To re-enable the CTK module execute the following command: sudo mv /Library/Frameworks/eToken.framework/Versions/A/SafeNet Authentication Client/Contents/PlugIns.disable /Library/Frameworks/eToken.framework/Versions/A/SafeNet Authentication Client/Contents/PlugIns
- Ensure that all used applications are supported with the Crypto Token Kit module.
- Reboot your machine after disabling/renaming the CTK module.

Uninstall

After SafeNet Authentication Client 10.1 Mac has been installed, it can be uninstalled. Local administrator rights are required to uninstall SafeNet Authentication Client. When SafeNet Authentication Client is uninstalled, user configuration and policy files may be deleted.

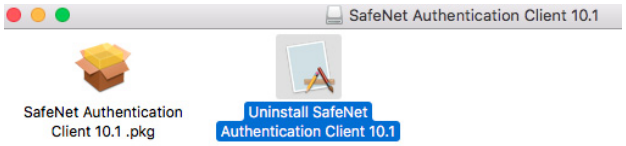
Uninstalling - Mac

Before uninstalling SafeNet Authentication Client 10.1 Mac, make sure that SafeNet Authentication Client Tools is closed.

To uninstall SafeNet Authentication Client 10.1 Mac:

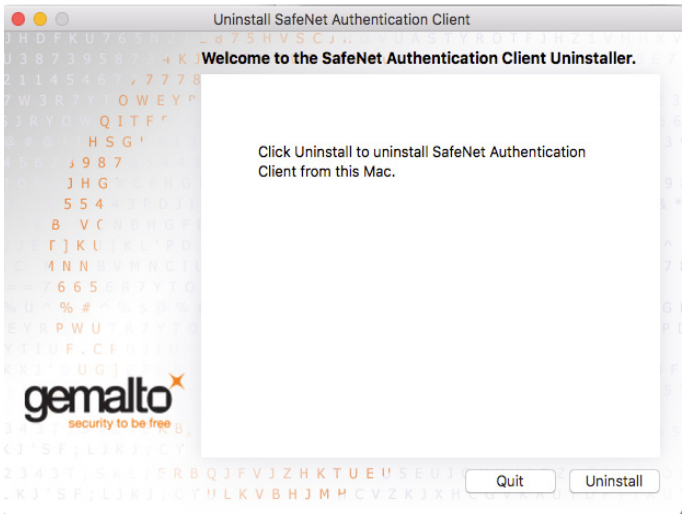
1. Double click **SafeNetAuthenticationClient.10.1.x.0.dmg** file.

   A new disk image file is created in the Finder window, including an mpkg installation file and an uninstall application.



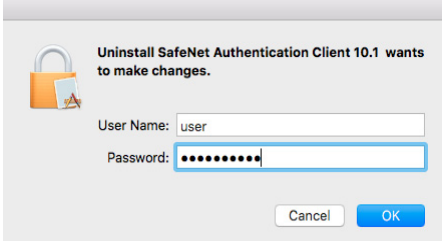
2. Click **Uninstall SafeNet Authentication Client (Mac) 10.1**.

   The **Welcome to the SafeNet Authentication Client Uninstaller** window opens.



3. Click **Uninstall**.

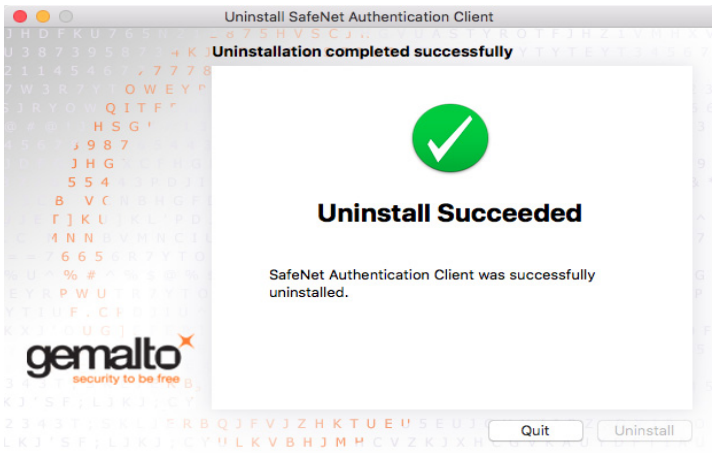
   The *Authenticate* window opens.



4. Enter a **Name and Password** and click **OK**.

> **NOTE**:
> You require Administrator permissions to uninstall SafeNet Authentication Client Mac 10.1.

The *Uninstallation completed successfully* window opens.



5. Click **Quite**.

Configuration Properties

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

NOTE:
All properties can be manually set and edited.

eToken Configuration Keys

SafeNet Virtual Token keys are located in `/etc/eToken.common.conf`.

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

Apple Keychain

Apple Keychain is Apple Computer's password management system in Mac OS X. Keychain Access is a Mac OS X application that allows the user to access the Apple Keychain and configure its contents.

SafeNet Authentication Client (Mac) provides a plug-in to support integration with Mac OS X Keychain Access. The plug-in is installed during SafeNet Authentication Client (Mac) installation.

Features Supported by Keychain Access

The SafeNet Authentication Client (Mac) Keychain Access integration supports the following features:

- Upload of certificates from the token to Keychain Access.
- Encryption and Decryption - by uploading certificates from a token to Keychain, they become available for applications, such as Mail, that can use the certificates to encrypt and decrypt mail messages.

Keychain Access Limitations

The following limitations apply when working with Keychain Access and SafeNet tokens.

- Keychain cannot be used to create new certificates. It can only upload certificates already located on the token.
- Change token password is not supported (however, it can be changed using SafeNet Authentication Client).
- It is not possible to import a certificate from a file to a token (however, certificates can be imported using SafeNet Authentication Client (Mac) Tools).
- The Keychain does not support RSA key generation to a token.

Displaying Token in Keychain Access

When you launch Keychain Access, you see a list of all the items in your Keychain, including information about each item's name, kind, creation date, and modification date.

When you insert a token, the device is displayed in the *Keychains* list.

To display token contents:

In the Keychains list on the left of the window, select token, then select an item from the Category list.

The details are displayed in the right section of the screen.

TIP

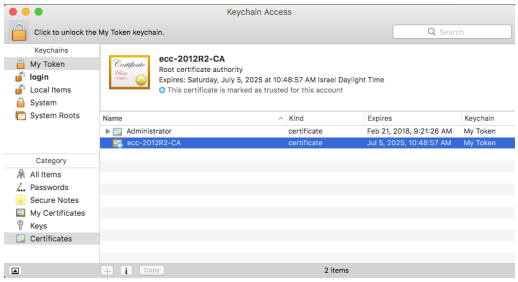
For details about performing additional functions with Keychain Access, refer to Mac OS X documentation.

Configuring Mac Keychain to Work with SSL and Secure Mail (S/MIME)

Mac Keychain must be configured to enable Safari to work with an SSL Connection and to enable encryption and decryption of emails.

To enable Mac Keychain to work with SSL and Secure Mail (S/MIME):

1. Open the *Keychain Access* window.



2. Double click on the root CA.

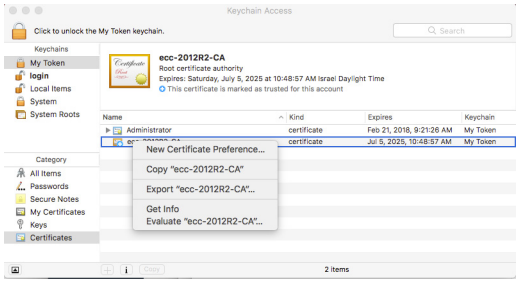
   The window with the certificate details opens.

3. Click on **Trust** to expand the section.
4. Set *Secure Socket Layer (SSL)* and/or *Secure Mail (S/MIME)* to **Always Trust.**
5. Close the window.

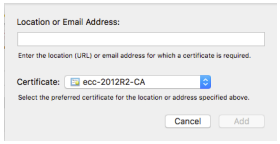
   You are returned to the Keychain Access window.

   The root CA certificate is now trusted for SSL and S/MIME operations.

6. Right-click on the Users Certificate and select **New Certificate Preferences**.



The *Location of Email Address* window opens.



7. In the Certificate field, select the required certificate.
8. Do one of the following and click **Add**:
   - For S/MIME, enter the email address of your mail account
   - For SSL, enter the URL of your secured site.

   The item is added to the *login* Keychain.

NOTE:
- You must configure SSL for each required secured website.
- SSL Authentication is not supported by Keychain when using Common Criteria devices in unlinked mode.

If you configured Secure email (S/MIME), you will now be prompted to enter the token password when signing and sending an email or when decrypting an encrypted email.

If you configured SSL for your secured sites, when logging on with Safari you will be prompted for the token password.

General Settings

The following settings are written to the appropriate section `/etc/eToken.conf/[General]`

NOTE:
On a Mac OS X, the number of slots is 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 SafeNet Authentication Client is backward compatible with Gemalto 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 via IDPrimePKCS11.dll For more information on Multi-Slots, see the **PKCS#11 Digital Signature PIN Authentication** section of the SafeNet Authentication Client User Guide.	**Value Name:** MultiSlotSupport Values: =0, =1 **1** - Multi-Slot support is enabled **0** - Multi-Slot support is disabled **Default**: 1
Software Slots Defines the number of virtual readers for SafeNet Virtual Tokens. **Note:** Can be modified in 'Reader Settings' in SafeNet Authentication Client Tools also. On Windows Vista 64-bit and on systems later than Windows 7 and Window 2008 R2, the total number of readers is limited to 10 from among: iKey readers, eToken readers, third-party readers, and reader emulations.	**Value Name:** SoftwareSlots **Values**: >=0 (0 = SafeNet Virtual Token is disabled; only physical tokens are enabled) **Default**: 2
PCSC Slots Defines the total number of PC/SC slots for all USB tokens and smartcards. Included in this total: • the number of allocated readers for third-party providers • the number of allocated iKey readers, which is defined during installation and cannot be changed • the number of allocated readers for other SafeNet physical tokens, which can be modified in 'Reader Settings' in SafeNet Authentication Client Tools **Note:** On Windows Vista 64-bit and on systems later than Windows 7 and Window 2008 R2, the total number of readers, consisting of this value and any enabled reader emulations, is limited to 10.	**Value Name:** PcscSlots **Values**: >=0 (0 = Physical tokens are disabled; only SafeNet Virtual Token is enabled) **Default**: 8
HID Slots Defines the total number of HID slots for all HID USB tokens.	**Value Name:** HIDSlots **Values**: =0, =4, >=0 **Default**: 4 slots

Description (Cont.)	Value (Cont.)
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 SafeNet Authentication Client 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 per 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 was not selected by default.	**Value Name:** TolerantX509Attributes **Values**: **1 (True)** - The attributes can differ **0 (False**) - Check that the values match **Default**: 0 (False)
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)
Disconnect SafeNet Virtual Token on Logoff Determines if SafeNet Virtual Tokens are disconnected when the user logs off.	**Value Name:** EtvLogoffUnplug **Values**: **1 (True)** - Disconnect SafeNet Virtual Token when logging off **0 (False)** - Do not disconnect SafeNet Virtual Token when logging off **Default**: 0 (False)

Description (Cont.)	Value (Cont.)
Protect Symmetric Keys Determines if symmetric keys are protected **Note:** If selected, even non-sensitive symmetric keys cannot be 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 were initialized as "eToken PKI Client 3.65 compatible" in SafeNet Authentication Client 8.0 and later, set this value to 0 to improve 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. To avoid having to authenticate every time a cryptographic operation is required for certificates containing Entrust certificate OID details, remove the default registration key value.	**Value Name:** NonRepudiationOID **Value**: Empty **Default**: No override
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)

Token-Domain Password Settings

The following settings are written to the appropriate section `/etc/eToken.conf/[SyncPin]`

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

License Settings

The following settings are written to the appropriate folder's `SafeNet\Authentication\SAC\License` registry key.

Description	Value
SAC License String Defines the license string issued by SafeNet for product registration	**Value Name:** License **Values**: License string provided by SafeNet **Default**: None

Initialization Settings

The following settings are written to the appropriate section `/etc/eToken.conf/[INIT]`

NOTE:
All setting in this section are not relevant to IDPrime MD 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

Description	Value (Cont.)
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
RSA-2048 Determines if the token support 2048-bit RSA keys by default. **Note:** Can be set in SafeNet Authentication Client Tools.	**Value Name:** RSA-2048 **Values**: **1(True)** - 2048-bit RSA keys are supported **0 (False)** - 2048-bit RSA keys are not supported **Default:** 0 (False)
OTP Support Determines if the token supports OTP generation by default. This setting enables HMAC-SHA1 support, required by OTP tokens. **Note:** Can be set in SafeNet Authentication Client Tools.	**Value Name:** HMAC-SHA1 Values: **1 (True)** - OTP generation is supported **0 (False)** - OTP generation is not supported **Default**: 1 (True), for OTP tokens. 0 (False), for other tokens
RSA Area Size For CardOS-based tokens, defines the default size, in bytes, of the area to reserve for RSA keys. • The size of the area allocated on the token is determined during token initialization, and cannot be modified without initializing the token. • RSA-Area-Size is not relevant when Legacy-Format-Version is set to 5. **Note:** Can be set in SafeNet Authentication Client Tools.	**Value Name:** RSA-Area-Size **Default:** depends on the token size: • For 16 K tokens, enough bytes for three 1024-bit keys • For 32 K tokens, enough bytes for five 1024-bit keys • For larger tokens, enough bytes for seven 1024-bit keys
Default Token Name Defines the default Token Name written to tokens during initialization.	Value Name: DefaultLabel **Value**: String **Default**: My Token

Description	Value (Cont.)
API: Keep Token Settings When initializing the token using the 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)
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 was 0 (False). As CardOS 4.2 does not support both FIPS and RSA-2048, failure to take this into account this 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 can be modified after initialization.	Value Name: PrvCachingModify Values: **1 (True) -** Can be modified **0 (False) -** Cannot be modified **Default**: 0 (False)
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)

Description	Value (Cont.)
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 appropriate section `/etc/eToken.conf/[AccessControl]`

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 appropriate section `/etc/eToken.conf/[InitApp]`

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 Token Initialization window. **Note:** This option is selected by default. If the option is de-selected, it can be selected again only by setting the registry key.	**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. **Note:** This option is not supported by iKey.	**Value Name:** MustChangePassword **Value**: **1** - Selected **0** - Not selected **Default**: 1
Private Data Caching If *Enable Private Cache* is selected, determines the token's private data cache default behavior. **Note:** Can be set in SafeNet Authentication Client Tools. This option is not supported by IDPrime MD 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 SafeNet Authentication Client Tools. This option is not supported by IDPrime MD 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**
RSA Secondary Authentication Mode (continued). **Note:** This option is not supported by IDPrime MD cards.	

Description (Cont.)	Value (Cont.)
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
Maximum number of 1024-bit RSA keys Defines the amount of space to reserve on the token for Common Criteria certificates that use 1024 -bit RSA keys. **Note:** This option is not supported by IDPrime MD cards.	**Value Name:** NumOfCertificatesWith1024Keys_help **Values**: 0-16 certificates **Default**: 0
Maximum number of 2048-bit RSA keys Defines the amount of space to reserve on the token for Common Criteria certificates that use 2048-bit RSA keys. **Note:** This option is not supported by IDPrime MD cards.	**Value Name:** NumOfCertificatesWith2048Keys_help **Values**: 1-16 certificates **Default**: 4

SafeNet Authentication Client Tools UI Settings

The following settings are written to the appropriate section `/etc/eToken.conf/[UI]`

Description	Value
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**: o (False)
Password Term Defines the term used for the token's user password.	**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 SafeNet Authentication Client 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

Description (Cont.)	Value (Cont.)
iKey LED On Determines when the connected iKey LED is on. **Note:** When working with applications related to Citrix, set this value to 0.	**Value Name:** IKeyLEDOn **Values**: **1** - The iKey LED is always on when SAC Monitor is running **0** -The iKey LED is on when the token has open connections only **Default**:1
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 SafeNet Authentication Client Tools	**Value Name:** HomeUrl **Values** (String): Valid URL **Default**: SafeNet's home URL
eToken Anywhere Determines if eToken Anywhere features are supported	**Value Name:** AnywhereExtendedMode **Values**: **1** -Supported **0** -Not supported **Default**: 1
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

Description (Cont.)	Value (Cont.)
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's "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 No default
Website URL If "Open website" is selected in the "Warning Message Click Action" setting, defines the URL to display	**Value Name:** ActionWebSiteURL **Values** (string): Website address **No default**

Description (Cont.)	Value (Cont.)
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 appropriate section `/etc/eToken.conf/[PQ]`

NOTE:
These settings are not relevant to IDPrime MD cards and eToken 5110 CC, as the password quality settings reside on the card itself.

Description	Value
Password - Minimum Length Defines the minimum password length. **Note:** Can be set in SafeNet Authentication Client Tools.	**Value Name:** pqMinLen **Values**: >=4 **Default**: 6
Password - Maximum Length Defines the maximum password length. **Note:** Can be set in SafeNet Authentication Client 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 SafeNet Authentication Client 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 SafeNet Authentication Client Tools. **Note:** Does not apply to iKey devices.	**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 SafeNet Authentication Client 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 SafeNet Authentication Client Tools.	**Value Name:** pqHistorySize **Values**: >= 0 (0 = No minimum) **Default**: 10 (iKey device history is limited to 6)

Description (Cont.)	Value (Cont.)
Password - Maximum Consecutive Repetitions Defines the maximum number of consecutive times a character can be used in a password. **Note:** Can be set in SafeNet Authentication Client Tools. **Note:** Does not apply to iKey devices.	**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 SafeNet Authentication Client Tools.	**Value Name:** pqMixChars **Values**: **1** - A minimum of 2 or 3 types must be included, as defined in the *Password- Minimum Mixed Character Type*s 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 - Complex*ity setting is set to **Standard complexity.** • Can be set in SafeNet Authentication Client 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 - Complex*ity setting is set to **Manual complexity.** • Can be set in SafeNet Authentication Client Tools.	**Value Name:** pqNumbers **Values**: 0 -Permitted 1 - Forbidden 2 - Mandatory **Default**: 0
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 SafeNet Authentication Client Tools.	**Value Name:** pqUpperCase **Values**: 0 - Permitted 1 - Forbidden 2 - Mandatory **Default**: 0

Description (Cont.)	Value (Cont.)
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 SafeNet Authentication Client 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 SafeNet Authentication Client 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:** We recommend that this policy 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
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. See the *Password Quality Owner* setting.	**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 appropriate section `/etc/eToken.conf/[AccessControl]`

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

The table lists all the *Access Control Features*.

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

Access Control Feature	Value Name	Description
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.
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.
Disconnect SafeNet Virtual Token	DisconnectVirtual	Enables/Disables the *Disconnect* SafeNet Virtual Token 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.
Connect SafeNet Virtual Token	AddeTokenVirtual	Enables/Disables the *Connect* SafeNet Virtual Token feature in SafeNet Authentication Client Tools.
Initialize Token	InitializeEToken	Enables/Disables the *Initialize Token* feature in SafeNet Authentication Client Tools.
Import Certificate	ImportCertificate	Enables/Disables the *Import Certificate* feature in SafeNet Authentication Client Tools.

Access Control Feature	Value Name (Cont.)	Description (Cont.)
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.
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.
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	CommonCriteriaPasswordSetting	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 - Generate OTP	GenerateOTP	Enables/Disables the *Generate OTP* 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**
System Tray -Change Token Password	TrayIconChangePassword	Enables/Disables the *Change Token Password* feature in the SafeNet Authentication Client Tray Menu.
System Tray - Select Token	SwitcheToken	Enables/Disables the *Select Token* feature in the SafeNet Authentication Client Tray Menu.

Access Control Feature	Value Name (Cont.)	Description (Cont.)
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	IdentrusChangePassword	Enables/Disables the *Change IdenTrust PIN* feature in SafeNet Authentication Client Tools.
Enable Unblock IdenTrust Passcode	IdentrusUnlock	Enables/Disables the *Unlock IdenTrust* feature in SafeNet Authentication Client Tools.
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.
Note: This property cannot be set in the Access Control Properties window. It must be set in the registry key.	VerisignSerialNumber	Enables/Disables the *Verisign Serial number* feature in SafeNet Authentication Client Tools.

Security Settings

The following settings are written to the appropriate section `/etc/eToken.conf/[Crypto]`

Description	Value
Key Management Defines key creation, export, unwrap, and off-board crypto policies.	**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 **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 • Disable any usage of symmetric keys off-board including unwrap **Default**: Compatible

Description (Cont.)	Value (Cont.)
Unsupported Cryptographic Algorithms and Features	**Value Name:** Disable-Crypto **Values**: (String) **None** - All algorithms are supported **Obsolete** - The following are disabled: MD5, RC2, RSA<1024, DES, GenericSecret<80, RC4<80, ECC<160, ECB, RSA-RAW **Manual** - Create your own list of algorithms The following can be disabled: **Algorithms:** RSA, ECC, AES, DES, 3DES, RC2, RC4, SHA2, SHA1, MD5, HMAC, GenericSecret **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<1024 **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-RSA-PSS-SHA2, AES-CTR, AES-MAC, Decrypt-RC2, Wrap-ECB" **Default**: None

SafeNet Authentication Client Security Enhancements

Enforcing Restrictive Cryptographic Policies

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

- Key Management Security Policy - See *Security Settings* on page 44 for more details.
- Disable Cryptographic Algorithm Policy - See *Security Settings* on page 44 for more details.

The motivation behind these enhancements:

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

  The following enhancements were 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.

  Once a restrictive policy has been set, the use of SafeNet Authentication Client with the above algorithms will be blocked. This might have implications on the way in which the third-party's applications currently work.

NOTE:
Administrators must make sure that the third-party applications used by the organization are configured accordingly and do no use one of the algorithms listed above, as they will be blocked.

Creating Symmetric Key Objects using PKCS#11

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

1. Protected memory was 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 were created with Secured Messaging (SM) protection so that the Microsoft smart card transport layer does not contain any APDU data with plain symmetric key material.

   For Secure Messaging (SM) to support the AES/3DES and Generic symmetric keys in SAC 10.2, 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.

   Keys that are created with previous SAC versions or on eToken Java devices which are formatted in non-FIPS/CC mode will not be 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).

Log Settings

The following settings are written to the appropriate section `/etc/eToken.conf/[Log]`

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

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.

Installing SAC using Gatekeeper

Gatekeeper makes it safer to download applications by protecting you from inadvertently installing malicious software on your Mac. Gatekeeper gives you more control over what you install.

Gatekeeper gives you three security options. You can download and install applications from anywhere on the web. Or you can choose the safest option and download and install applications only from the Mac App Store. To install SAC using GateKeeper use the following Link:

https://serviceportal.safenet-inc.com/eservice_ENU/start.swe?SWECmd=Start&SWEHo=serviceportal.safenet-inc.com

Gatekeeper helps OS X makes sure that application is safe to run, even if it's not from the Mac App Store. Gatekeeper checks for the presence of a digital certificate embedded in the application itself that tells the Mac the application is from a signed developer who has registered with Apple

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.

Additional Recommendations

While no system can be 100 percent immune from every threat, Mac OS lets you do the following to keep your information as safe as possible:

- Turn on a firewall to prevent other machines from accessing services running on your Mac.
- Control access to your Mac by locking your screen after a period of inactivity.
- Set up secure file sharing.
- Use Password Assistant to create stronger passwords for local utilities like Users & Groups.
- Make sure you're only running sharing services that you really need.