



Administrator Guide for eToken RTE 3.65

January 2006



Contact Information

Support

If you have any questions regarding this package, its documentation and content or how to obtain a valid software license you may contact your local reseller or Aladdin's technical support team:

Country / Region	Telephone
USA	1-212-329-6658 1-800-223-3494
EUROPE: Austria, Belgium, France, Germany, Netherlands, Spain, Switzerland, UK	00800-22523346
Ireland	0011800-22523346
Rest of the World	+972-3-6362266 ext 2

If you want to write to the eToken Technical Support department, please go to the following web page:

http://www.Aladdin.com/forms/eToken_question/form.asp

Website

<http://www.Aladdin.com/eToken>

COPYRIGHTS AND TRADEMARKS

The eToken™ system and its documentation are copyrighted © 1985 to present, by Aladdin Knowledge Systems Ltd.

All rights reserved.

eToken™ is a trademark and ALADDIN KNOWLEDGE SYSTEMS LTD is a registered trademark of Aladdin Knowledge Systems Ltd.

All other trademarks, brands, and product names used in this guide are trademarks of their respective owners.

This manual and the information contained herein are confidential and proprietary to Aladdin Knowledge Systems Ltd. (hereinafter "Aladdin"). All intellectual property rights (including, without limitation, copyrights, trade secrets, trademarks, etc.) evidenced by or embodied in and/or attached/connected/related to this manual, information contained herein and the Product, are and shall be owned solely by Aladdin. Aladdin does not convey to you an interest in or to this manual, information contained herein and the Product, but only a limited right of use. Any unauthorized use, disclosure or reproduction is a violation of the licenses and/or Aladdin's proprietary rights and will be prosecuted to the full extent of the Law.

NOTICE

All attempts have been made to make the information in this document complete and accurate. Aladdin is not responsible for any direct or indirect damages or loss of business resulting from inaccuracies or omissions. The specifications in this document are subject to change without notice.

ALADDIN KNOWLEDGE SYSTEMS LTD.**eTOKEN ENTERPRISE END USER LICENSE AGREEMENT**

IMPORTANT INFORMATION - PLEASE READ THIS AGREEMENT CAREFULLY BEFORE OPENING THE PACKAGE AND/OR USING THE CONTENTS THEREOF AND/OR BEFORE DOWNLOADING OR INSTALLING THE SOFTWARE PROGRAM. ALL ORDERS FOR AND USE OF THE eTOKEN ENTERPRISE PRODUCTS (including without limitation, libraries, utilities, diskettes, CD-ROM, eToken™ keys and the accompanying technical documentation) (hereinafter "Product") SUPPLIED BY ALADDIN KNOWLEDGE SYSTEMS LTD. (or any of its affiliates - either of them referred to as "ALADDIN") ARE AND SHALL BE, SUBJECT TO THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT. BY OPENING THE PACKAGE CONTAINING THE PRODUCTS AND/OR BY DOWNLOADING THE SOFTWARE (as defined hereunder) AND/OR BY INSTALLING THE SOFTWARE ON YOUR COMPUTER AND/OR BY USING THE PRODUCT, YOU ARE ACCEPTING THIS AGREEMENT AND AGREEING TO BE BOUND BY ITS TERMS AND CONDITIONS.

IF YOU DO NOT AGREE TO THIS AGREEMENT DO NOT OPEN THE PACKAGE AND/OR DOWNLOAD AND/OR INSTALL THE SOFTWARE AND PROMPTLY (within 7 days from the date you received this package) RETURN THE PRODUCTS WITH THE ORIGINAL PACKAGE AND THE PROOF OF PAYMENT TO ALADDIN, ERASE THE SOFTWARE, AND ANY PART THEREOF, FROM YOUR COMPUTER AND DO NOT USE IT IN ANY MANNER WHATSOEVER.

1. **Title & Ownership.** The object code version of the software component of Aladdin's eToken Enterprise Product, including any revisions, corrections, modifications, enhancements, updates and/or upgrades thereto about to be installed by you, (hereinafter in whole or any part thereof defined as: "**Software**"), and the related documentation, ARE NOT FOR SALE and are and shall remain in Aladdin's sole property. All intellectual property rights (including, without limitation, copyrights, trade secrets, trademarks, etc.) evidenced by or embodied in and/or attached/connected/related to the Product, are and shall be owned solely by Aladdin. This Agreement does not convey to you an interest in or to the Software, but only a limited right of use revocable in accordance with the terms of this Agreement. Nothing in this Agreement constitutes a waiver of Aladdin's intellectual property rights under any law.
2. **License.** Subject to payment of applicable fees, Aladdin hereby grants to you, and you accept, a personal, nonexclusive and fully revocable limited License to use the Software, in executable form only, as described in the Software accompanying technical documentation and only according to the terms of this Agreement: (i) you may install the Software and use it on computers located in your place of business, as described in Aladdin's related documentation; and (ii) you may merge and link the Software into your computer programs for the sole purpose described in the accompanying technical guide provided by Aladdin ("**Technical Guide**").
3. **Prohibited Uses.** The Product must be used and maintained in strict compliance with the instruction and safety precautions of Aladdin contained herein, in all supplements thereto and in any other written documents of Aladdin. Except as specifically permitted in Sections 1 and 2 above, you agree not to (i) use, modify, merge or sub-license the Software or any other of Aladdin's Products, except as expressly authorized in this Agreement and in the Technical Guide; and (ii) sell, license (or sub-license), lease, assign, transfer, pledge, or share your rights under this License with/to anyone else; and (iii) modify, disassemble, decompile, reverse engineer, revise or enhance the Software or attempt to discover the Software's source code; and (iv) place the Software onto a server so that it is accessible via a public network; and (v) use any back-up or archival copies of the Software (or allow someone else to use such copies) for any purpose other than to replace an original copy if it is destroyed or becomes defective. If you are a member of the European Union, this agreement does not affect your rights under any legislation implementing the EC Council Directive on the Legal Protection of Computer Programs. If you seek any information within the meaning of that Directive you should initially approach Aladdin.

4. **Maintenance and Support.** Aladdin has no obligation to provide support, maintenance, upgrades, modifications, or new releases under this Agreement.
5. **Limited Warranty.** Aladdin warrants, for your benefit alone, that (i) the Software, when and as delivered to you, and for a period of three (3) months after the date of delivery to you, will perform in substantial compliance with the Technical Guide, provided that it is used on the computer hardware and with the operating system for which it was designed; and (ii) that the eToken™ key, for a period of twelve (12) months after the date of delivery to you, will be substantially free from significant defects in materials and workmanship.
6. **Warranty Disclaimer.** ALADDIN DOES NOT WARRANT THAT ANY OF ITS PRODUCT(S) WILL MEET YOUR REQUIREMENTS OR THAT ITS OPERATION WILL BE UNINTERRUPTED OR ERROR-FREE. TO THE EXTENT ALLOWED BY LAW, ALADDIN EXPRESSLY DISCLAIMS ALL EXPRESS WARRANTIES NOT STATED HEREIN AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NO ALADDIN'S DEALER, DISTRIBUTOR, RESELLER, AGENT OR EMPLOYEE IS AUTHORIZED TO MAKE ANY MODIFICATIONS, EXTENSIONS, OR ADDITIONS TO THIS WARRANTY. If any modifications are made to the Software or to any other part of the Product by you during the warranty period; if the media and the eToken™ key is subjected to accident, abuse, or improper use; the Product has not been properly installed, operated, repaired or maintained in accordance with the instructions supplied by Aladdin; the Product has been subjected to abnormal physical or electrical stress, negligence or accident; or if you violate any of the terms of this Agreement, then the warranty in Section 5 above, shall immediately be terminated. The warranty shall not apply if the Software is used on or in conjunction with hardware or program other than the unmodified version of hardware and program with which the Software was designed to be used as described in the Technical Guide.
7. **Limitation of Remedies.** In the event of a breach of this warranty, Aladdin's sole obligation shall be, at Aladdin's sole discretion: (i) to replace or repair the Product, or component thereof, that does not meet the foregoing limited warranty, free of charge; (ii) to refund the price paid by you for the Product, or component thereof. Any replacement or repaired component will be warranted for the remainder of the original warranty period or 30 days, whichever is longer. Warranty claims must be made in writing during the warranty period and within seven (7) days of the observation of the defect accompanied by evidence satisfactory to Aladdin. All Products should be returned to the distributor from which they were purchased (if not purchased directly from Aladdin) and shall be shipped by the returning party with freight and insurance paid. The Product or component thereof must be returned with a copy of your receipt.
8. **Exclusion Of Consequential Damages.** The parties acknowledge that Product is inherently complex and may not be completely free of errors. ALADDIN SHALL NOT BE LIABLE (WHETHER UNDER CONTRACT, TORT (INCLUDING NEGLIGENCE) OR OTHERWISE) TO YOU, OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE (INCLUDING INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES), INCLUDING, WITHOUT LIMITATION, ANY LOSS OR DAMAGE TO BUSINESS EARNINGS, LOST PROFITS OR GOODWILL AND LOST OR DAMAGED DATA OR DOCUMENTATION, SUFFERED BY ANY PERSON, ARISING FROM AND/OR RELATED WITH AND/OR CONNECTED TO DELIVERY, INSTALLATION, USE OR PERFORMANCE OF THE PRODUCT AND/OR ANY COMPONENT OF THE PRODUCT, EVEN IF ALADDIN IS ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
9. **Limitation Of Liability.** IN THE EVENT THAT, NOTWITHSTANDING THE TERMS OF THIS AGREEMENT, ALADDIN IS FOUND LIABLE FOR DAMAGES BASED ON ANY DEFECT OR NONCONFORMITY OF ITS PRODUCT(S), ITS TOTAL LIABILITY FOR EACH DEFECTIVE PRODUCT SHALL NOT EXCEED THE PRICE PAID TO ALADDIN FOR SUCH PRODUCT.
10. **Termination.** Your failure to comply with the terms of this Agreement shall terminate your license and this Agreement. Upon termination of this Agreement: (i) the License granted to you in this Agreement shall expire and you, upon termination, shall discontinue all further

use of the Software and other licensed Product(s); and (ii) you shall promptly return to Aladdin all tangible property representing Aladdin's intellectual property rights and all copies thereof and/or shall erase/delete any such information held by it in electronic form. Sections 1, 3, 6-11 shall survive any termination of this Agreement.

11. **Governing Law & Jurisdiction.** This Agreement shall be construed and governed in accordance with the laws of Israel (except for conflict of law provisions) and only the courts in Israel shall have jurisdiction in any conflict or dispute arising out of this Agreement. The application of the United Nations Convention of Contracts for the International Sale of Goods is expressly excluded. The failure of either party to enforce any rights granted hereunder or to take action against the other party in the event of any breach hereunder shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches.
12. **Government Regulation and Export Control.** You agree that the Product will not be shipped, transferred, or exported into any country or used in any manner prohibited by applicable law. It is stipulated that the Product is subject to certain export control laws, rules, and/or regulations, including, without limiting the foregoing, to the United States and/or Israeli export control laws, rules, and/or regulations. You undertake to comply in all respects with the export and re-export restriction as set forth herein and any update made thereto from time to time.
13. **Third Party Software.** Product contains third party software, as set forth in Exhibit A. Such third party's software is provided "As Is" and use of such software shall be governed by the terms and conditions as set forth in Exhibit A. If the Product contains any software provided by third parties other than the software noted in Exhibit A, such third party's software are provided "As Is" and shall be subject to the terms of the provisions and condition set forth in the agreements contained/attached to such software. In the event such agreements are not available, such third party software shall be provided "As Is" without any warranty of any kind and Sections 2, 3, 6, 8, 9-12 of this Agreement shall apply to all such third party software providers and third party software as if they were Aladdin and the Product respectively.
14. **Miscellaneous.** This Agreement represents the complete agreement concerning this License and may be amended only by a written agreement executed by both parties. If any provision of this Agreement is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

I HAVE READ AND UNDERSTOOD THIS AGREEMENT AND AGREE TO BE BOUND BY ALL OF THE TERMS.

Exhibit A**A. Notices.**

- I.** Product has incorporated source code licensed under the Mozilla Public License ("MPL").
- II.** MPL is available at <http://www.mozilla.org/MPL/>

The MPL License, version 1.1, Copyright © 1998-2004 The Mozilla Organization.

- III.** The source code is freely available from:
<http://lxr.mozilla.org/mozilla/source/security/nss/cmd/modutil/modutil.c/>
- IV.** "Covered Code" means: source code governed by the MPL.

B. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

FCC Compliance

eToken USB has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation.

This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:

- a.Reorient or relocate the receiving antenna.
- b.Increase the separation between the equipment and receiver.
- c.Connect the equipment to an outlet on a circuit different from that to which the receiver is connected.
- d.Consult the dealer or an experienced radio/TV technician.

FCC Warning

Modifications not expressly approved by the manufacturer could void the user authority to operate the equipment under FCC rules.

All of the above applies also to the eToken USB.

FCC authorities have determined that the rest of the eToken product line does not contain a Class B Computing Device Peripheral and therefore does not require FCC regulation.

CE Compliance



The eToken product line complies with the CE EMC Directive and related standards*. eToken products are marked with the CE logo and an eToken CE conformity card is included in every shipment or upon demand.

*EMC directive 89/336/EEC and related standards EN 55022, EN 50082-1.

UL Certification

The eToken product line successfully completed UL 94 Tests for Flammability of Plastic Materials for Parts in Devices and Appliances. eToken products comply with UL 1950 Safety of Information Technology Equipment regulations.

ISO 9002 Certification



The eToken product line is designed and manufactured by Aladdin Knowledge Systems, an ISO 9002-certified company. Aladdin's quality assurance system is approved by the International Organization for Standardization (ISO), ensuring that Aladdin products and customer service standards consistently meet specifications in order to provide outstanding customer satisfaction.

Certificate of Compliance

Upon request, Aladdin Knowledge Systems will supply a Certificate of Compliance to any software developer who wishes to demonstrate that the eToken product line conforms to the specifications stated. Software developers can distribute this certificate to the end user along with their programs.

Table of Contents

Chapter 1	1
Introduction	1
RTE Overview	2
What's New in eToken RTE 3.65	3
Minimum Requirements	4
Chapter 2	5
Getting Started	5
Local User installation	6
Enabling your eToken	7
Starting eToken Properties	7
Connecting the eToken Extension Cable	9
Chapter 3	11
Main Administration Tasks	11
Setting Up a New eToken User	12
Issuing a Replacement eToken	13
Recovering eTokens from Employees	14
Administrator Password	14
Initializing eToken	15
eToken Password Quality	16
Chapter 4	17
eToken Deployment	17
Deploying eToken RTE in the Organization	18
Configuring eToken Properties	19
RTE Installation - Command Line Options	21
Registry Settings	23
Chapter 5	31
RTE Attributes	31
What is a Certificate Store	32

eToken Certificate Store	33
Modes of Operation.....	35
Propagation Mode.....	35
TokenView Mode	35
Importing PFX files.....	37
PFX Troubleshooting	39
Computer Standby Behavior.....	41
Password Retry Counters	41
User Interface Policy.....	42
RTE Backwards Compatibility	44
Chapter 6.....	45
eToken Properties	45
Local Machine Configuration Options	46
Certificate Store Options.....	47
CA Certificate Loading	49
Readers Management.....	50
Power Saving Options.....	51
General Control Buttons	52
eToken Configuration Options	53
Basic eToken Properties	53
Changing the eToken Password.....	56
Renaming the eToken.....	58
Unblocking the eToken	60
Advanced eToken Properties.....	63
Details tab	65
Settings tab	66
Certificates & Keys tab.....	69
Administrator tab	78
Setting the eToken Administrator Password.....	80
Setting the eToken User Password	81
Chapter 7	83
eToken Initialization.....	83

Initializing the eToken	84
Initializing using Customizable Parameters.....	87
Setting or Changing the Initialization Key	92
Multi-token Initialization	94
Stopping the Initialization Process.....	96
Password Reset.....	96
Troubleshooting	97
Chapter 8.....	99
eToken Password Quality	99
How Password Quality Works	100
Calculating the Password Quality Value.....	100
Change Password Policy.....	101
Password Quality Parameters	104
Editing the Password Quality Configuration File	108
Using Your Own Configuration File	111
Chapter 9.....	113
Troubleshooting.....	113
Problems and Possible Solutions	114
Checking USB Support	116
Technical Support.....	118

Chapter 1

Introduction

The eToken RTE 3.65 has new features and additional functionality than previous versions. An introduction to the RTE is provided in this chapter.

About This Chapter

This chapter contains the following sections:

- ◆ “RTE Overview”, on page 2, presents a short description of the RTE and what it does.
- ◆ “What’s New in eToken RTE 3.65”, on page 3, provides a brief explanation of new features, additions and changes to the RTE from previous versions.
- ◆ “Minimum Requirements”, on page 4 lists the hardware, software and operating system requirements for using eToken.

RTE Overview

The eToken Run Time Environment (RTE) installs all the necessary files and eToken drivers to support eToken integration with various security applications. It enables Windows operating systems and third party applications to access the eToken. Installing the RTE allows communication with all available eToken devices and forms the basis for Aladdin's various security solutions. These include eToken PKI solutions using either PKCS#11 or CAPI, proprietary eToken applications such as WSO (Web Sign-On), SSO (Simple Sign-On), eToken for Network Logon and management solutions like eToken TMS – a Token Management System that is a complete framework for managing all aspects of token assignment, deployment and personalization within an organization.

Aladdin's eToken PKI Solutions enable the implementation of strong two-factor authentication using standard certificates. Generic integration with both Microsoft CAPI and PKCS#11 security interfaces enables interoperability with a variety of security application such As Web Access, VPN Access, Network Logon, PC Protection and Secure eMail. PKI keys and certificates can be securely created, stored and used from within the eToken.

When used with eToken PRO / Smartcard or eToken NG-OTP the PKI Private keys are generated and operate on board the secure chip.

eToken RTE supports the various types of eToken devices in both form factors. This means that only a single RTE installation is required to enable operations of either a traditional Smartcard or a USB Token (PRO,NG-OTP or R2), and results in easy deployment and cost effective installation in use of eToken products and solutions.

eToken RTE can be deployed and updated using any standard software distribution system such as SMS. In addition, the eToken Management System (TMS) supports software distribution using the Microsoft GPO system.

What's New in eToken RTE 3.65

The eToken RTE 3.65 is enhanced with new features and additional functionality from previous versions of the eToken RTE.

The new eToken RTE 3.65:

- ◆ Provides support for new tokens:
 - eToken devices with the CardOS 4.20B and 4.30B operating systems. These will include versions of the eToken PRO (32K and 64K) and eToken NG-OTP (32K and 64K).
These new tokens provide the user with better performance, more EEPROM and support for RSA 2048-bit cryptography.
- ◆ Provides better support in CAPI-enabled applications for keys and certificates that were created by PKCS#11-enabled applications.
- ◆ Supports the Supplementary API (SAPI) to eToken as introduced in SDK 3.60.
- ◆ Provides better integration of PKCS#11 keys through CAPI.

Minimum Requirements

The following are the minimum requirements for using eToken:

- ◆ PC with at least 10 MB disk space.
- ◆ Windows 2000 (with Service Pack 4 or later installed) or Windows XP with full functionality of all new features.
- ◆ Windows 98, Windows NT 4.0 (with Service Pack 6 or later installed), Windows Me. The administrator should be aware that certain elements of the new functionality are **not** supported on these platforms, like the ability to import a PFX file from eToken Properties.
- ◆ Microsoft Windows Installer (MSI) 1.1 or later.
Internet Explorer 5.0 or later. MSI 1.1 is included with all installations of Windows 2000, Windows Me and Windows XP.
For details, please see: www.Aladdin.com/etoken
- ◆ At least one USB port, with USB support enabled in the BIOS.

NOTE:

Additional software may be required for individual eToken solutions.
For more information, please refer to www.Aladdin.com/etoken.

Chapter 2

Getting Started

This chapter provides the basic information you need to start using eToken, and gives detailed instructions for installing and personalizing eToken.

About This Chapter

The chapter includes the following sections:

- ◆ “Local User installation”, on page 6, details how a local user should install the RTE.
- ◆ “Enabling your eToken”, on page 7, explains the procedure for starting to use the RTE.
- ◆ “Connecting the eToken Extension Cable”, on page 9, explains how to connect the eToken extension cable if needed.

Local User installation

Users installing the eToken RTE individually should use the RTE msi file or installation CD Rom which will launch the Installation Wizard. For details on installing the RTE with this Wizard, please see the eToken User Guide.

Administrators wanting to install the eToken RTE throughout the organization with specific parameters should use the command line installation method as described in Deploying eToken RTE in the Organization RTE Installation - Command Line Options on page 21.

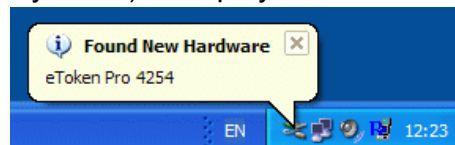
For detailed instructions for installing, integrating and using specific eToken solutions, please refer to www.Aladdin.com/etoken

Enabling your eToken

After installing the RTE, it is necessary to enable the eToken the first time it is inserted into the USB port.

➤ **To enable the eToken:**

- 1 Insert your eToken into the USB port or alternatively the USB extension cable for the first time. The eToken lights up and during this process, which may take a few moments, the **Found New Hardware** pop-up on the Start Bar (The image displayed is from a Windows XP system. The hardware recognition steps and messages may vary on other Operating Systems) is displayed:



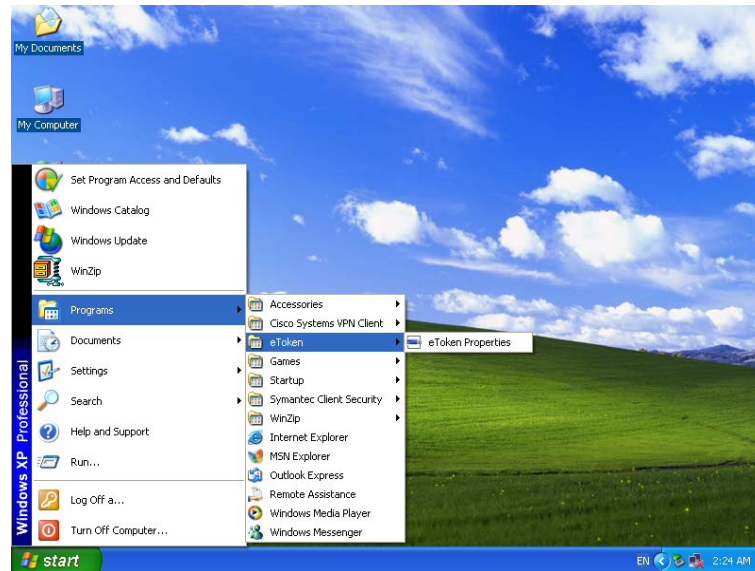
- 2 The hardware installation continues until complete when the eToken is ready to be used.

Starting eToken Properties

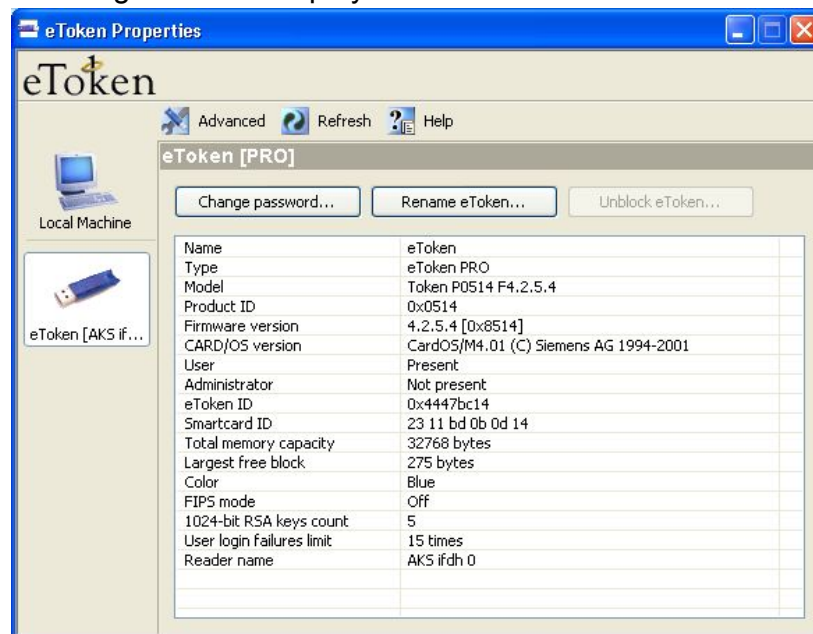
To enable and use your eToken with eToken Properties you must first start eToken Properties.

➤ **To start eToken Properties:**

- 1 From the **Start** menu, select **Programs >eToken >eToken Properties** and the following is displayed:



- 2 Click **eToken Properties** and with your eToken inserted, the following screen is displayed:



- 3 You are now ready to work with eToken Properties.

Connecting the eToken Extension Cable

The eToken connects to the computer's USB port. If the USB port is located at the back of the PC, it is probably difficult to reach. The eToken extension cable enables easy access to the USB port for insertion and removal of the eToken. Extension cables are available from your local Aladdin dealer.

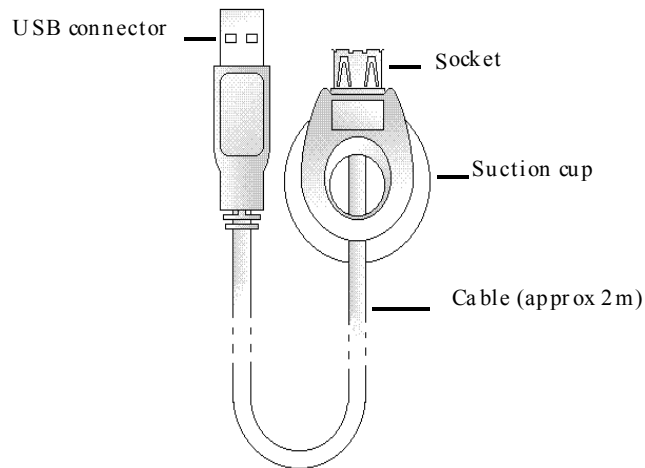
If a USB port or hub is located on the keyboard or monitor, you may not need an eToken extension cable. If the port is on the monitor, make sure that the monitor is connected to the USB port of the PC through a standard USB type A to type B cable.

Your eToken extension cable package includes:

- ◆ A round, translucent sticker.
- ◆ A cable, two meters (approximately six linear feet) long, with USB type A to type B connectors.

At one end of the cable is a socket and a special suction cup. This end should be mounted in a convenient place, so that you can easily insert and remove the eToken. At the other end of the cable is a small plug that connects to the existing USB connector on the PC.

USBconnector plug



➤ **To install the eToken extension cable:**

- 1 Locate the computer's USB port, and insert the small USB connector plug into it.
- 2 Peel off the sticker and paste it in a convenient place, for example, on the side of the monitor or on the casing of the PC.
- 3 Affix the suction cup of the eToken extension cable to the smooth surface of the sticker, pressing it firmly in place.
- 4 Plug the eToken into the cable socket and make sure it lights up.

Chapter 3

Main Administration Tasks

Administering eToken in an organization is simple and straightforward. This chapter details the main administration tasks required for ongoing use of eTokens within the organization.

About This Chapter

The chapter includes the following sections:

- ◆ "Setting Up a New eToken User", on page 12, outlines what needs to be done to set up a new user with eToken.
- ◆ "Registry Settings", on page 23 explains how to manipulate certain registry keys.
- ◆ "Issuing a Replacement eToken", on page 13, explains what actions to take in the event of damaged, lost or stolen, or forgotten eToken passwords.
- ◆ "Recovering eTokens from Employees", on page 14, explains what to do when eToken users leave the organization.
- ◆ "Initializing eToken", on page 15 provides general information on the process of initializing an eToken.
- ◆ "eToken Password Quality", on page 16 explains the various settings used in amending the Password policy.

Setting Up a New eToken User

When a new employee joins the organization, do the following:

- ◆ Install the eToken RTE on the employee's computer.
- ◆ If required, install any additional installation for the relevant eToken solution.
- ◆ Issue the employee a new eToken, with the instructions for personalizing it. This includes "Changing the eToken Password" on page 56, "Renaming the eToken" on page 58, and "Unblocking the eToken" on page 60.

In eToken RTE 3.00 and prior versions, the Administrator password functionality was limited to opening a blocked user password and enabling the initializing of FIPS tokens.

Since eToken RTE 3.51 the Administrator password is also used to protect certain configuration files.

NOTE for eToken initialized using RTE 3.00

The first time an administrator uses the eToken (e.g. login/change password), a few of the password attributes are changed according to the new RTE functionality. Among these changes is that the Error Retry counter is automatically set to 15 times, regardless of the actual Initialization setting.

Issuing a Replacement eToken

A user's eToken may need to be replaced if the eToken is lost or damaged. When a user reports a lost or damaged eToken, you should discard the eToken and issue the user another eToken, with a requirement to personalize it as soon as possible.

If the user forgets the eToken password there are two different solutions depending on eToken device is being employed.

In the eToken R2, if a user forgets the password for his or her personal eToken, it cannot be used for any eToken-based operation. The eToken password is stored securely on the eToken, and it is not possible to reset it or replace it.

In this case the procedure for dealing with a forgotten eToken password is exactly the same as for a lost or damaged eToken.

In the eToken PRO and eToken NG-OTP, if a user forgets the eToken password, the eToken can either be reinitialized whereby the token's details are erased and the token is reset to the default password, or the user password can be reset using the system administrator password with all of the token's details preserved.

Recovering eTokens from Employees

When an employee leaves the organization, in addition to taking the standard actions, such as revoking any current certificates and closing network accounts, you should recover his or her eToken and its password. You can then choose to discard the eToken, or to change the current password and reuse the eToken.

If you are unable to recover the eToken password from the employee, the eToken is unusable and you should discard it if you do not have an administrator password on it. An eToken can be used only on computers that have been set up for use with specific eToken Enterprise security applications.

Administrator Password

A special Administrator password function is included with the RTE. This function enables administrators, even if they do not know the user password, to use the eToken to recover and use information on the eToken. If a user forgets or loses his password, the administrator can recover the necessary information from the eToken.

If this functionality is required, the administrator **MUST** first initialize the eToken with this administrator password before distributing the eTokens throughout the organization.

Initializing eToken

The eToken Initialization feature erases all data on an eToken PRO or eToken NG-OTP and resets the file structure according to various configurable parameters. In addition the feature can set the Administrator password and other functional parameters.

The eToken PRO can be initialized as a standard eToken PRO, as a FIPS eToken PRO and as a blank eToken.

For detailed information on the eToken Initialization feature included in the eToken Properties configuration tool, please refer to Chapter 7: Initializing the eToken on page 83 of this Administrator Guide.

eToken Password Quality

Altering the password changing policy is controlled by the *etpass.ini* file in the OS system directory. The eToken RTE installation installs an *etpass.ini* file if not already present. If such a file is present, the existing file remains intact and is not changed.

However an IT administrator can define different settings for this file and replace the existing *etpass.ini* file on any user's machine.

The *etpass.ini* file controls the password changing policy when using an eToken application. When using other (i.e. non eToken) application mechanisms to change the eToken password (e.g. changing the password via Netscape) these settings are not relevant.

The eToken Password Quality feature allows the administrator to edit the password quality parameters to suit the organization's password policy. For detailed information on the eToken Password Quality feature included in the eToken Properties configuration tool, please refer to Chapter 8 about the eToken Password Quality feature on page 99 of this Administrator Guide.

Chapter 4

eToken Deployment

Deploying eToken in an organization is simple and straightforward. This chapter provides administration guidelines in respect of the Administrator's role in defining the RTE 3.65 features to be used and how to manipulate these features.

About This Chapter

The chapter includes the following sections:

- ◆ "Deploying eToken RTE in the Organization", on page 18, details how to deploy eToken RTE in the organization and the various deployment options available.
- ◆ "Registry Settings", on page 23, details the available registry settings and what they mean.

Deploying eToken RTE in the Organization

The eToken runtime environment (RTE) includes all the necessary files and drivers to support eToken integration. It also includes the eToken Properties configuration tool, which enables easy user management of the eToken password and name.

The eToken RTE (version 3.65) must be installed on each computer on which eToken is to be used.

The RTE can be installed with or without the eToken Properties configuration tool. When installing for the first time and you want to install all components including eToken Initialization, you must run the installation from the command line as follows:

```
msiexec /i RTE_3.65.msi
```

This installs the RTE **AND** the eToken Properties configuration tool. If eToken Properties is not required, it can be removed by running the following command line:

```
msiexec /i RTE_3.65.msi REMOVE=eTProps /qb
```

Alternatively you can install just the RTE module. This loads all the PKI interfaces (CAPI and PKCS#11) and the necessary eToken drivers.

```
msiexec /i RTE_3.65.msi ADDDEFAULT=rteFeature /qb
```

qb is optional and means “run in silent mode”.

NOTE:

When uninstalling an RTE version, the Registry settings are **NOT** removed. As a result when installing a new version of the RTE, the previous registry settings will remain active unless explicitly changed with new command line settings.

Configuring eToken Properties

The eToken Properties configuration tool now includes additional features. These are:

- ◆ **Advanced Mode** – Enables/Disables “Local Machine” and “Advanced” buttons.
- ◆ **Initialize individual token** – shows or hides the “Initialization” button (top toolbar).
- ◆ **Multi-token Initialization** – (simultaneous initialize number of tokens) – shows or hides the “Multi-token Initialization” button (Local Machine panel).
- ◆ **Password Quality** – shows or hides the “Password Quality” button (Local Machine panel).
- ◆ **Set AUX Key Container** - shows or hides the “AUX Key” button on the cryptography panel (Certificates & keys tab, see page 69).

The administrator should run the installation from the command line as follows:

```
msiexec /i RTE_3.65.msi ETPROPS_MODE = <value>
```

The RTE default installation only includes **Advanced mode** and the msi runs with the value 1.

The administrator can install eToken Properties with a different configuration based on the features to be installed. The available configuration options (<value> can be combination of numbers) are:

Advanced Mode	1 (ALWAYS)
Initialize individual token	2
Multi-token Initialization	4
Password Quality	8
Set AUX Key Container	16

In all cases, the value “1” must be included, since the Advanced mode enables the other eToken Properties options to work.

Examples:

Complete Installation:

"Msiexec /i RTE_3.65.msi ETPROPS_MODE = 31"

Means <Value>: 31 = 1+2+4+8+16 (Decimal values)

This will result in the features being installed as follows:

- Advanced Mode YES
- Initialize individual token YES
- Multi-token Initialization YES
- Password Quality YES
- Set AUX Key Container YES

Installation with Initialization and Password Quality:

"Msiexec /i RTE_3.65.msi ETPROPS_MODE = 11"

Means <Value>: 11 = 1+2+8 (Decimal values)

This will result in the features being installed as follows:

- Advanced Mode YES
- Initialize individual token YES
- Multi-token Initialization NO
- Password Quality YES
- Set AUX Key Container NO

RTE Installation - Command Line Options

An IT Administrator can set some of the RTE characteristics during the installation process on the user's machine.

The following settings are available during Installation:

ETPROPS_MODE:

0 disable mode, 1 advanced mode (default).

For other configuration options please see Configuring eToken Properties on page 19.

LOAD_LOCAL:

0 TokenView mode, 1 Propagation mode (default).

AUTO_DUPLICATION_CHECK:

0 Disable (default), 1 Auto Check Enabled.

FRIENDLY_NAME_VER:

1 (default) or 2.

(Relevant only when Load Local is off, i.e. set to 0)

CA_CERT_MODE:

0 Prompt User (Default), 1 Automatic, 2 Never.

ET_PROCLISTMODE_X:

Empty (Default)

ET_UI_POLICY:

0 (Default)

For information on the available options, please see User Interface Policy on page 42.

NO_SMARTCARD_LOGON_PIN_DLG:

0 disable (default), 1 Suppress token removal report on standby

ET_UI_PWD_TIMEOUT:

Empty (Default)

SLOW_NET:

Empty (Default)

READER_COUNT:

Set to n where n is the number of readers attached to the computer.

This option enables the administrator to set the number of virtual readers during installation. There is NO equivalent registry key for this option.

For more details on the command line entries above, please see Registry Settings , on page 23.

Installation Command Line samples

```
msiexec /i RTE_3.65.msi ETPROPS_MODE=0 AUTO_DUPLICATION_CHECK=1 /qb
```

This command will install the RTE with a basic Installation User Interface (only a progress bar is shown: /qb) and with eTProperties Advanced mode disabled, and with automatic key duplication test on. All other settings are set to their default values.

```
msiexec /i RTE_3.65.msi /q
```

This command will install the RTE in a silent mode with all installation properties set to their defaults.

NOTE:

If reboot is needed in a silent mode it will be activated automatically.

```
msiexec /x RTE_3.65.msi /q
```

This command line should uninstall the RTE in a silent mode.

Registry Settings

An IT administrator can control the behavior of some of the RTE features installed on a user machine by manipulating Registry Keys. Some of these keys can be set via the installation process (see RTE Installation - Command Line Options, page 21).

NOTE:

It is up to the IT administrator to set the security attributes of these registry keys - so that a user will not be able to change them.

All the Registry settings can be found under the following path:

[HKEY_LOCAL_MACHINE\SOFTWARE\Aladdin\Token](#)

eTProperties Advanced Mode

An administrator is able to disable the Advanced mode of the **eTProperties** application. When Advanced mode is disabled (by using the value 0 in the command line installation), both the **Local Machine** and **Advanced** buttons are disabled.

The Advanced mode is controlled by a registry key. This option can be set during command line installation of the RTE (see RTE Installation - Command Line Options, page 21).

Registry Key:

[HKEY_LOCAL_MACHINE\SOFTWARE\Aladdin\Token\ETProperties](#)

Value:

[DWORD Advanced](#) (Default Value = 1)

Command Line Option: ETPROPS_MODE

For more information see Local Machine Configuration Options on page 46.

eTProperties Additional Logos

An administrator can define a right side image for eTProperties. This option is set by a registry key.

Registry Key:

[HKEY_LOCAL_MACHINE\SOFTWARE\Aladdin\Token\Token\Properties](#)

Value:

[STRING Logo](#)

Should have the path to a BMP file.
The bitmap size is: Height<=33, Width<=512.

[DWORD LogoTransparency](#)

When you want to use PIXEL (0,0) of the bitmap as a transparent color set the [LogoTransparency](#) value to a non-zero value e.g. 1.

Certificate Visibility

This option controls behavior of the eToken Certificate Store. It allows the user to select between Propagation Mode and TokenView mode. This option has some effect on Certificate's related operations. This option is controlled by a registry key.

Registry Key:

[HKEY_LOCAL_MACHINE\SOFTWARE\Aladdin\Token\TCertStore](#)

Value:

[DWORD LoadLocal](#) (Default Value = 1)

Command Line Option: LOAD_LOCAL

If set to 1 the Propagation mode is activated and if set to 0 TokenView mode is activated. This option can be set during command line installation of the RTE (see RTE Installation - Command Line Options, page 21).

For further information, see Certificate Store Options, on page 47.

Key Duplication Test

In different situations, keys that are being imported to the token might be found on the Host machine (managed by MS CSP). For security reasons, eToken keys should not be stored on the local machine as well. **eToken CertStore** can perform an automatic duplication key test during Certificate Enumeration. This test has an overhead that affects performance - and could be significant if there are many keys in the registry. The Automatic Key Duplication Test is controlled by a registry key.

Registry Key:

[HKEY_LOCAL_MACHINE\SOFTWARE\Aladdin\eToken\eTCertStore](#)

Value:

[DWORD DuplicateKeyTest](#) (Default Value = 0)

Command Line Option: AUTO_DUPPLICATION_CHECK

If value is set to 0 - No automatic test will be performed. This registry key is also controlled by eTProperties under Local Machine section. In addition, the key can be set during command line installation of the RTE (see RTE Installation - Command Line Options, page 21).

eToken Certificate Friendly Name Version

eToken Certificate Store (eTCertStore) sets the Friendly Name of a certificate in two possible formats:

Format 1:

The reader is at the beginning followed by the Certificate's subject.

Example: AKS ifdh 0:: john brown...

Format 2:

The certificate subject, followed by its usage, followed by the reader name.

Example: john brown: Server Authentication, Client

Authentication, Code Signing, Secure Email, Time Stamping...

reader::AKS ifdh 0.

The flavor of the Friendly Name is controlled by the registry key.

Registry Key:

HKEY_LOCAL_MACHINE\SOFTWARE\Aladdin\Token\TCertStore

Value:

DWORD FriendlyNameVer (Default Value = 1)

Command Line Option: FRIENDLY_NAME_VER

By setting FriendlyNameVer to 2 the format of the Friendly Name will be as described in Format 2 above. This option can be set during command line installation of the RTE (see RTE Installation - Command Line Options, page 21).

For further information, see Modes of Operation on page 35, and Certificate Store Options, on page 47.

CA Certificate Handling

An eToken with CA certificates on it may be inserted into the computer. It may occur that a particular CA certificate is not on the computer and this registry setting defines what to do in such a case.

Registry Key:

HKEY_LOCAL_MACHINE\SOFTWARE\Aladdin\Token\TCertStore\

HKEY_CURRENT_USER\SOFTWARE\Aladdin\Token\TCertStore\

Value:

DWORD CACertMode (Default Value = 0)

Value Options: 0 = Prompt before installing CA certificate

1 = Automatically install CA certificate

2 = Do not install CA certificate

Command Line Option: CA_CERT_MODE

If there is a need to make a change, ensure that the change is made only to the current user key:

HKEY_CURRENT_USER\SOFTWARE\Aladdin\Token\TCertStore\

Please see CA Certificate Loading on page 49 for more information.

Suppress Certificate Propagation

In some applications (e.g. eTProperties) where we need to manipulate the certificates on the token, the Propagation mode is not wanted. To turn it off for a specific process, enter the name of the process under the following key in the registry. The name of the process must be without the .exe extension and must be closed by a " ; ". Example: eTProps;

Registry Key:

[HKEY_LOCAL_MACHINE\SOFTWARE\Aladdin\eToken\eTCertStore\](#)

Value:

[STRING ProcLoadLocalIgnore.](#)

(Default Value = eTProps; AppViewer; capiView; ckView; eTCertConv)

By adding a process to the list, eToken Certificate Store will not look for eToken certificates in the registry during certificate enumerations.

CAPI Suppress Password Caching

Under normal conditions, each time a process uses private data from the eToken or needs the password to proceed, the user will be asked to enter it. The password is normally cached during a particular session so that multiple password presentations can be avoided. If password caching is suppressed, the user will be asked to enter the password each time it is required.

Registry Key:

[HKEY_LOCAL_MACHINE\SOFTWARE\Aladdin\eToken\eTCAPI\](#)

Value:

[STRING ProcListModeX](#) (Default Value is EMPTY)

Command Line Option: ET_PROCLISTMODE_X

CAPI User Interface Policy

This deals with non-standard behavior as in the case of a Winlogon password. The key controls the functioning of behavior in respect of unblocking a locked eToken or a forced password change.

Registry Key:

[HKEY_LOCAL_MACHINE\SOFTWARE\Aladdin\Token\TCAPI\](#)

Value:

[DWORD UI_Policy](#) (Default Value = 0)

Command Line Option: ET_UI_POLICY

For more information see User Interface Policy on page 42.

CAPI Timeout Logon Settings

This allows the setting of the timeout period for logon.

Many CAPI applications (such as MS Outlook or IE) do not log in to the token explicitly. When they access the token for the first time, the pop-up window in which to enter the token password appears. The application then remains logged in forever (or until the token is removed). This registry entry allows setting the timeout value (in seconds). After passing that time, the password should be re-entered.

Registry Key:

[HKEY_LOCAL_MACHINE\SOFTWARE\Aladdin\Token\TCAPI\](#)

Value:

[DWORD PasswordTimeout](#) (No Default Value created)

Command Line Option: ET_UI_PWD_TIMEOUT

Remote Behavior Customization

This allows customization of RTE remote behavior. Working via Terminal Services, the RTE behaves differently in the case of fast and slow (dial-up) connections. In the second case the token is available in the read-only mode.

By default, the RTE automatically detects whether the connection seems to be too slow (and provides read-only functionality in that case). This registry key allows setting the mode explicitly. Set 0 to disable 'slow network' mode or a positive value to enforce it.

Registry Key:

[HKEY_LOCAL_MACHINE\SOFTWARE\Aladdin\Token\Core](#)

Value:

[DWORD SlowNet](#) (No Default Value created)

Command Line Option: SLOW_NET

Standby Behavior

After returning from Standby\Hibernate mode, this setting indicates whether the token removal and insertion will be reported by the driver.

Registry key:

[HKEY_LOCAL_MACHINE\SOFTWARE\Aladdin\Token\Core\](#) (DWORD)

Value:

[DWORD NoSmartcardLogonPinDlg](#) (No Default Value created)

Command Line Option: NO_SMARTCARD_LOGON_PIN_DLG

The absence of this entry or a zero value means the token removal/insertion is reported while a non-zero value in this entry means no report will be made.

For further information, see Computer Standby Behavior, on page 41.

Chapter 5

RTE Attributes

This chapter deals with various aspects of how the RTE operates as well as the policies and practises behind the RTE. Its attributes are explained for the administrator to understand and be able to work effectively with them.

About This Chapter

The chapter includes the following sections:

- ◆ “What is a Certificate Store”, on page 32, explains the basic concept of a Certificate Store.
- ◆ “eToken Certificate Store”, on page 33 details how the eToken Certificate Store operates.
- ◆ “Modes of Operation”, on page 35, provides information on operating in the Propagation and TokenView modes.
- ◆ “Computer Standby Behavior” on page 41, explains how the RTE handles the computer in standby mode.
- ◆ “Password Retry Counters” on page 41, explains this feature and how it works.
- ◆ “User Interface Policy” on page 42, defines and explains certain non-standard behavior in certain applications.
- ◆ “RTE Backwards Compatibility”, on page 44, highlights the issues concerning the use of previous RTE versions.

What is a Certificate Store

A certificate store (CS) is a concept for managing and controlling a collection of certificates. There are many such collections on a computer: one keeps personal certificates, a different one keeps certificates from trusted authorities, etc.

Certificates belonging to the user are located in the store named **Personal**. Developers may know this store by the name **My**. This store has no specific physical location - it is rather a logical store - a collection of one or more physical stores that behave as a single store for the user. All certificates located in these physical stores are visible as if they were located in one store called **Personal**. Any attempt to delete one certificate will delete this certificate from the corresponding physical store (where that particular certificate is located) and so on.

Normally, there is only one physical store in the Personal CS: Registry.

eToken Certificate Store

The eToken RTE adds one further physical store to the Personal collection: **eToken store** which represents the certificates stored on the tokens themselves. However, it is possible to store certificates on the token without dealing with Certificate Store.

Applications may use CAPI to store the certificate as an “attribute” of the key (CryptSetKeyParam with KP_CERTIFICATE). In fact, all enrollment programs (i.e. programs that generate certificate requests and generate certificates) use this method regardless of writing the certificate to the certificate store.

Aladdin recommends all developers use this mechanism to store certificates. If you use this mechanism you must take into account the following:

- ◆ You may not store a certificate without having the key on the token (the eToken Certificate Store does not allow this in any case). This is always the case unless the certificate is a Root CA certificate, in which case it will be installed on the token even though the certificate does not have a private key.
- ◆ You may store only one certificate per key

Certificates can also be read by using CAPI (CryptGetKeyParam with KP_CERTIFICATE), but very few applications actually reach certificates in this manner. Usually these are smartcard-aware applications like WinLogon or complicated applications with rich security features like Entrust 7.0.

Most applications, such as Outlook or Internet Explorer, use certificate stores. Therefore your certificates need to be represented in a certificate store in order to use them. They can be placed there in one of two ways:

- ◆ RTE may implement a certificate store to represent certificates located on tokens.
- ◆ RTE may propagate certificates from tokens to the Registry store (propagation mode).

Both modes are implemented in RTE 3.65.

For some security operations such as mail encryption and signature verification, it is sufficient to only possess a certificate. However to decrypt data sent to you or digitally sign data, in addition to a certificate you need a private key as well.

Applications operate with certificates because to a user the binary value of a private key is meaningless. Certificates possess certain printable information that allows users to distinguish between several different certificates. The private key associated with the certificate is needed to perform the security operation. The certificate contains a special property that points to a particular key and this association via the certificate property is crucial for the successful use of certificates. Generally, properties such as a friendly name or reference to a particular key are not parts of the X.509 certificate. They are not signed and therefore may be changed by the application.

In order to better understand the behavior of eToken RTE in relation to Certificate Store, the user should be aware that:

- ◆ There is a logical gap between the application and the RTE. RTE functions are not called directly by an application. The application calls Operating System functions such as CAPI or Certificate Store functions and as a result some RTE functions are called. It is not always possible to understand why and how the particular call occurred. For instance, an application may intend to write a certificate to the eToken Certificate Store or to the Personal Certificate Store. Since the eToken Certificate Store is part of the Personal collection, the same RTE function will be called with the same parameters - it is impossible to say which action was intended.
- ◆ All tokens are represented as a single store. This is because it is not possible to register separate stores for each token. Therefore, some situations cannot be handled properly. For instance, several tokens may contain the same certificate. However the certificate store cannot keep two identical certificates, so only one certificate (arbitrarily chosen by the RTE) will be represented.

Modes of Operation

eToken RTE 3.65 supports two different modes of operation as detailed below:

- ◆ Propagation Mode
- ◆ TokenView Mode

Propagation Mode

The work situation where **LoadLocal = 1**.

This is the default mode for working with the RTE and is described as follows:

- ◆ When some application opens the certificate store, the certificates from tokens are copied to the Registry store. For each token this operation is performed only once after inserting the token. All sub-sequential accesses to the Certificate Store do not touch this token. However, if the token is reinserted, the operation will be repeated.
- ◆ Since eToken propagates all certificates to the Registry store, it never gets any requests for certificate deletion. The certificate is deleted from the Registry store but remains on the token. Thus, on the next token insertion it will be copied again. You may use the eToken Properties configuration tool to remove certificates from the token.

The purpose of Propagation mode is not to give the best view of the currently inserted tokens via the certificate store, but to allow as painless an operation as possible for applications.

TokenView Mode

The work situation where **LoadLocal = 0**.

In this mode the eToken Certificate Store represents certificates on currently inserted tokens. This behavior may be briefly summarized as follows:

- ◆ When an application opens the eToken store, either directly or as part of the Personal collection, all certificates located on currently selected tokens are read. If there are identical certificates on several inserted tokens (i.e. the same X.509 encoded value) only one of them will be visible.
- ◆ If there is an identical certificate in the Registry store, it will not be represented. (This occurs in the case of a Smartcard Logon certificate that is read by Winlogon upon card insertion).
- ◆ The RTE will build a friendly name containing key usage and reader name.
- ◆ Writing of a certificate will be allowed only if there is corresponding key.
- ◆ During certificate deletion a special dialog box appears, that prompts the user about deletion of the key container. This occurs since normally keys are not removed by Microsoft. It may be acceptable for Microsoft CSP that keeps data on the PC, but not for eToken CSP, since the space on the token is very restricted.

Exceptions

Most applications work with both modes, but certain applications may not work properly when in TokenView mode. Some examples are:

- ◆ Certain applications under Windows XP (e.g. Outlook XP, Outlook Express): These applications open the certificate store only once, when the application is launched. If the eToken was not inserted when the application was launched, its certificates will not be visible.
- ◆ Entrust 7.0: Digital ID Monitor expects that once a user's certificate is found in the certificate store, all certificates belonging to the same profile will be available. Since Winlogon copies the Smartcard Logon certificate once you enter the token, it is not uncommon to see this certificate in the Registry store. So, Digital ID Monitor will expect to find other certificates of the same user, even if the token is currently not inserted.

Importing PFX files

A PFX (or P12) file is a file containing a private key, its certificate and optionally, the certificate chain up to the CA certificate. The PFX file is usually imported to the token (or computer) via the Certificate Import Wizard, which may be initiated in several ways:

- ◆ From the Internet Explorer Menu (Tools\Options).
- ◆ By right-clicking the Internet Explorer icon.
- ◆ By double-clicking a file with a .PFX or .P12 extension.
- ◆ From the MMC.

When the Certificate Import Wizard is launched, the following actions are performed by Microsoft:

- ◆ The file is decrypted with the password supplied by the user.
- ◆ The private key is created with parameters defined by the user (such as Exportable). Unless a particular provider has been explicitly specified in the PFX file, Microsoft CSP is used for this purpose.
- ◆ Certificates stored in the PFX file are stored in one or more certificate stores according to user selection. If the user selected a particular Certificate Store (logical or physical) it will serve as the destination for all certificates. Otherwise (i.e. automatic selection of certificate store), the Operating System will pass each certificate to the proper store:
 - A Private key's certificate will be written to the personal store.
 - A CA certificate (if present) will be written to the Trusted Root Certification Authorities. Other certificates of the chain (rare cases) will be written to the Intermediate Certification Authorities.

This presents a number of problems:

- ◆ There is no way to get the key directly to the eToken. It is the Operating System's decision as where to put the key - and unless the PFX file was exported from an eToken R2, the key is placed in the Microsoft CSP.

- ◆ The first (and only) point where the RTE may take control is when certificates are written. However, as explained before, eToken Certificate Store usually declines to write a certificate if there is no private key on the token. In this case there will be no private key since it was imported by the Microsoft CSP.
- ◆ If Automatic selection is used, a failure returned by the RTE is not propagated to the user. The Operating System just writes the certificate to the Registry store instead and states that the import was successful, so the user is confused.

As a result of these issues, Aladdin recommends importing PFX files in the following manner:

- ◆ Start the process by double-clicking the PFX/P12 file. RTE has a special way of handling PFX importing in order to export the missing private key from the Microsoft CSP. However, in order to do this, the RTE needs to recognize that the application is about to import the PFX file. The RTE is not able to do this if you initiate the process in any other way (IE, MMC).
- ◆ You should mark the key as Exportable.
- ◆ Select the physical store: My\eToken as a destination store for the certificates.
- ◆ A dialog box appears prompting you for a password. Pressing ESC is equivalent to importing a certificate to the machine. If you want the key and certificate to be on the token, select the token and enter the password.
- ◆ If the PFX file contains a CA certificate, you will be prompted. To put the CA certificate on the token, click **Yes**. In both cases the certificate will be put on the computer as well.

You may choose Automatic selection as well, but in this case you should take into account several things:

- ◆ If the process fails, an error message appears. It should be noted that the Operating System might still store the certificate on the computer.

- ◆ The key and certificate may not be imported to the token if there already was such certificate in the Personal store. This may happen, for instance, if you exported a PFX file on the same computer.

PFX Troubleshooting

Certain situations may arise that appear strange and require explanation.

Question:

If I insert two tokens with the same keys and certificates, how I can choose which one will be used for cryptographic operations?

Answer:

You cannot. It is impossible to present two similar certificates in the certificate store. The order of reading tokens is arbitrary. Do not do it.

Question:

I don't use propagation mode. Still, I see one of my certificates in the store when no token is inserted. Why?

Answer:

Winlogon propagates the Smartcard Logon certificate in any case.

Also, the default for Load Local is 1, which means that the certificate may have been loaded to the computer before disabling Load Local. After setting Load Local to 0 simply delete the certificate from the local machine and it will not appear again without an eToken being inserted.

Question:

In propagation mode, I export a PFX file from an eToken R2 and import it to an eToken PRO (or another eToken R2). Only the key is written, but the certificate is missing.

Answer:

Since you are working in propagation mode, the certificate was in the Registry.

During the import of the PFX file, the new key container will be created, but the certificate remains the same (there may not be two similar certificates in the store). Therefore, the key is not found.

Delete the certificate from the Registry store before importing PFX.

Question:

In propagation mode, I import a PFX file exported from R2 to another token. I select eToken Certificate Store explicitly. As a result, both key and certificate are imported successfully, regardless that the old certificate was in Registry store. So, why not work in this way?

Answer:

If you will try to import a PFX file containing a CA certificate, it will fail. If you know for certain that your PFX file does not contain a CA certificate, using the eToken certificate store is OK. But this cannot be recommended in general cases.

Computer Standby Behavior

In old versions of the RTE (3.51 and lower), when the computer went into standby mode or hibernation and was then powered up again, nothing was reported to the Smartcard Resource Manager to indicate that the smartcard had lost its state and therefore could still be used without a new login.

This meant that certain applications could still be used without having to re-authenticate the user even though the token was “removed and then reinserted”.

Since RTE 3.60, this approach can still be used, but the default is that after hibernation or standby mode, the driver reports to the Smartcard Resource Manager about reinsertion of the token. As a result the user will need to authenticate to the smartcard once more.

See Registry Settings: Standby Behavior on page 23 for more details.

Password Retry Counters

The password retry counter defines how many unsuccessful password attempts are allowed until the eToken is locked. The default setting is 15 tries.

RTE 3.65 shows a warning message when the number of remaining attempts is low, so that the user can be prevented from possibly locking the token.

It should be noted that this is a usability feature and not a security feature. The real number of remaining attempts is managed by the smartcard internally. This actual number cannot be seen or compromised by any malicious code.

The number of successful or unsuccessful password attempts for applications built using SDK 2.60 will not change the counter. As a result, users may be confused, but need to remember that there is no security implication in this event. Further, users should be aware that the warning message may appear even if the token is already locked.

This feature only works if the eToken was initialized using RTE 3.60 and later.

User Interface Policy

There is certain User Interface non-standard behavior that occurs with certain applications like Windows Network Logon and needs to be dealt with in a specific manner. Three kinds of behavior exist:

- ◆ **Unblocking of a blocked password:** This can only work when using an eToken initialized with an administrator password. If the administrator password has been enabled then a blocked password can be unblocked by following the Unblocking the eToken procedure as described on page 60. The user will receive a message explaining what the situation is. Then the Unblock user password dialog opens. Finally a message opens explaining that the original password presentation will fail but the user can use the new password.
- ◆ **Unblocking in the case of a wrong password:** This option works like the previous one - the only difference being that it is initiated after entering an incorrect password and not when the password is blocked. This is because, in most cases, entering the incorrect password is only the result of mistyping the password. It is recommended to keep this feature disabled.**Password change policies:** In this situation, the user must change the password before continuing to use the token.

The registry key: DWORD UI_Policy controls the functioning of this behavior.

The value is built from three hexadecimal digits (**XYZ**) where the **position** of the digit indicates which feature is being specified and the **value** of the digit indicates to which applications this feature applies.

Therefore:

- X** Stands for password change policies
- Y** Stands for unblocking in the case of a wrong password
- Z** Stands for unblocking of a blocked password

For each digit (X, Y or Z) you should choose one of the following:

- 0** Disables the option
- 1** Enables the option only for Windows Network logon
- 2** Enables the option for all CAPI application except Windows Network logon
- 3** Enables the option for all CAPI applications

Example

In order to apply password change policies for all applications but only allow the unblocking of a blocked password in Windows Network Logon, one should use the hexadecimal value 301, where:

- X = 3** Enables password change policies for all applications
- Y = 0** Disables unblocking in the case of a wrong password
- Z = 1** Enables the unblocking of a blocked password in Windows Network Logon only

It is strongly recommended to avoid using any UI features of the RTE (such as unblocking of the blocked token) from Windows Logon when working under Windows 2000. It may freeze the computer.

See Registry Settings CAPI User Interface Policy on page 28 for more details.

RTE Backwards Compatibility

- ◆ Please note that some details of Propagation mode may change in future versions of the RTE. As stated previously in Propagation mode the certificate store does not represent the picture of tokens.
- ◆ Implementation of the RTE was changed to accommodate new functionality and provide better support for Windows 98/Me. This results in broken binary compatibility with SDK 2.60. Consequently, this may create a deadlock situation if applications developed with SDK 2.60 mix both low-level (API) and high-level (Capi and/or PKCS#11) functions.

Aladdin is currently unaware of any applications with this possible conflict. Aladdin recommends that developers consider rebuilding applications using SDK 3.00.

Chapter 6

eToken Properties

eToken Properties provide administrators with a configuration tool to administer and set token policies. This tool enables users to perform basic token management such as password changes, viewing of information, and viewing of certificates on the eToken. In addition, eToken Properties provides users and administrators with a quick and easy way to transfer digital certificates and keys between a computer and an eToken.

eToken Properties also includes an initialization feature (See Chapter 7) allowing administrators to initialize the eToken according to specific organizational requirements or security modes and a password quality feature (See Chapter 8) enabling the manipulation of the parameters which calculate an eToken's password quality rating.

About This Chapter

This chapter provides a brief explanation of eToken Properties and the various configuration options available to the administrator and user respectively.

The chapter includes the following sections:

- ◆ “Local Machine Configuration Options”, on page 46, details the specific options available on the Local machine at all times.
- ◆ “eToken Configuration Options”, on page 53 explains all the configuration options available in both Basic and Advanced mode as well as setting of administrator and user passwords.

Local Machine Configuration Options

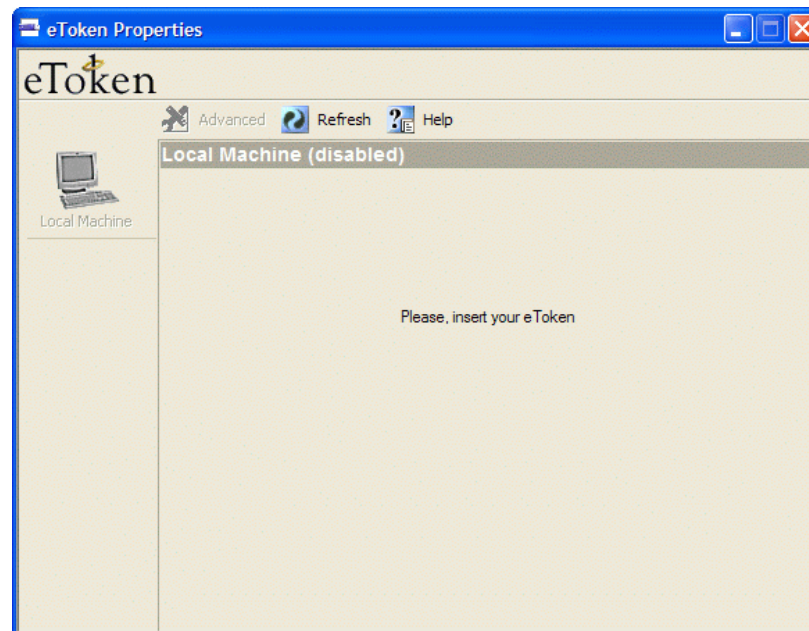
The Local Machine configuration options enable setting global parameters that affect the eToken operation. These options are displayed when **eToken Properties** is launched (and no eToken is inserted) or when the **Local Machine** button in the left panel is clicked, as shown below:



The window consists of a left vertical panel containing buttons and a right pane that contains information on the currently selected button.

The top button in the left panel is the **Local Machine** button, which is automatically selected when launching **eToken Properties**. The configuration options associated with this button are **not** specific to one eToken, but are general configuration options applicable to any eToken.

Some administrators may choose to disable **Local Machine** and **Advanced** features. In such a case the following is displayed:



In order to use the application in this configuration you will need to insert an eToken.

Certificate Store Options

Copy certificates to computer store

Default - enabled

PKI operations usually require certificates, private and public keys. Private keys should always be securely stored on the eToken. Certificates should also be stored on the eToken as this enables mobility (the certificate will be readily available when using the eToken on a different machine). See Modes of Operation, on page 35.

Since certificates themselves do not contain private information, selecting this box enables pre-loading of certificates from the eToken and caching them on the local machine. This considerably speeds up accessing of these certificates by various applications, and can dramatically shorten response time when several certificates on the eToken need to be enumerated by an application.

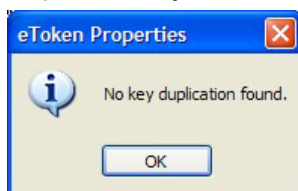
Auto key duplication check

Default - disabled

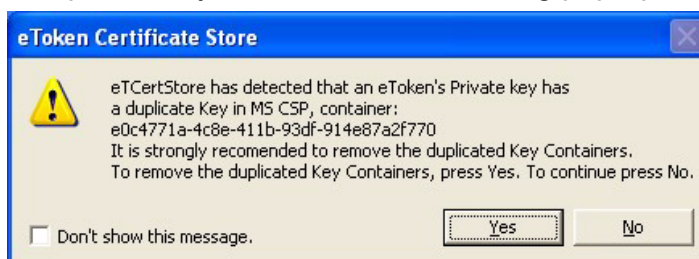
It is possible that private keys have, in the past, been placed on the computer. This leads to duplication in that there is a key on the computer AND on the eToken. For effective security, only one private key should be allowed. This key should always be kept on the eToken in order to maximize security.

To perform an automatic key duplication check each time an application enumerates the eToken certificates, select the **Auto key duplication check** check box. Note that this might slow down certificate and key operations.

Alternatively, if you want to check whether the Auto Key is duplicated on an ad hoc basis, click **Check key duplication**. If no duplicate keys are found, the following pop-up is displayed:



If duplicate keys are found, the following pop-up is displayed:



Click **Yes** to remove the duplicate keys from the computer.

CA Certificate Loading

CA certificates can be downloaded onto the eToken. When this eToken is inserted into the computer, one or more of these CA certificates may not be on the computer. In such a case, an option exists to load the CA certificate if desired.

When the RTE is first installed, it copies settings for the local machine to a Current User account and then works with the data from the current user account

The available options are:

Automatically install CA certificate

The CA certificate is copied to the computer without asking the user.

Prompt before installing CA certificate (Default)

When a CA certificate is to be installed, a message asking the user whether or not to copy the certificate is first displayed. Click **Yes** to copy or **No** not to copy the CA Certificate

Note:

The message box also has a check box "Don't ask again". Selecting this option will change the selected option above.

If you click **Yes**, the option changes to **Automatically install CA certificate**.

If you click **No**, the option changes to **Do not install CA certificate**.

Do not install CA certificate

The CA certificate is not installed at all.

Despite the settings chosen, it is possible that another dialog box from Microsoft opens asking if you wish to continue this action. This is standard Microsoft operating procedure because the action to be undertaken may affect security matters on the computer. If you want to copy the CA certificate, click **Yes** in this case.

Readers Management

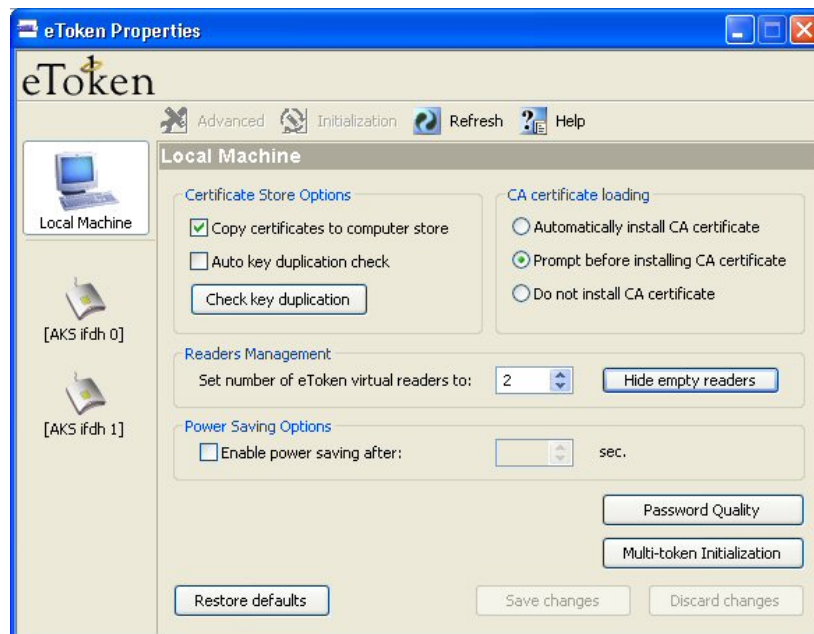
Set number of eToken virtual readers to:

Default - 2 readers

eToken RTE setup installs two virtual readers. This means two eTokens can be recognized at the same time and accessed by applications using them.

You can change the number of installed readers by changing the value of this field and thereby increase or decrease the number of eTokens that can be recognized simultaneously by the system.

The **Show empty readers** button is a toggle button that allows you to see what readers are installed on the system. When you click this button, the Local Machine left panel changes as displayed:



Below the **Local Machine** button (left panel), are buttons which represent eTokens and/or smartcard readers available on the system. When installing the eToken RTE, two virtual smartcard readers are installed with it. The names of eToken smartcard readers begin with **AKS ifdh**. This is followed by the reader number.

When an eToken is inserted into the USB port, it has the effect of inserting a smartcard into one of the readers. The button's icon changes to an eToken icon to reflect this.

Physical smartcard readers are also displayed if installed. Once a smartcard is inserted into these readers, the reader icon will change to one with a smartcard inserted as displayed:



Power Saving Options

Enable power saving after: ----Sec.

Default - disabled

Microsoft's Windows XP using an Intel processor has built in support for both USB 1.1 and USB 2.0 and incorporates support for USB "Selective Suspend". This feature allows the USB device driver which supports selective suspend to turn off the USB device it controls when the device is idle. In effect, the feature stops the USB host controller (HC) from polling if all ports are suspended and allows the processor to go to C3/C4 state. When the device is no longer idle and is to be used again, the system wakes the device and resumes normal operation. This option is particularly important when using portable devices (laptops, etc.).

C3/C4 states are low power states for the processor in which the processor saves power and under typical use conditions allows for battery life to be extended by~10%.

When the **Enable power saving after...** button is selected, you have the option to change how many seconds before the power saving mode activates.

In order to activate the change made to the power saving configuration, you need to remove the eToken and then reinsert it.

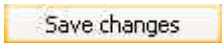
General Control Buttons

Restore defaults

A rectangular button with a yellow border and a light gray background, containing the text "Restore defaults".

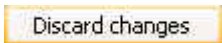
Clicking this button restores the local machine default configuration values.

Save changes

A rectangular button with a yellow border and a light gray background, containing the text "Save changes".

Clicking this button saves any changes that have been made to the local machine configuration values.

Discard changes

A rectangular button with a yellow border and a light gray background, containing the text "Discard changes".

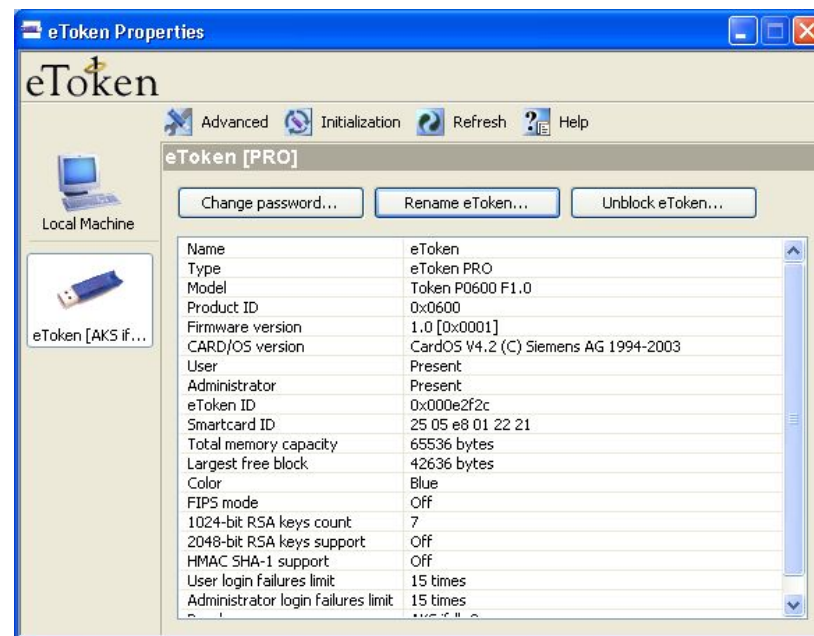
Clicking this button discards any changes that have been made to the local machine configuration values.

eToken Configuration Options

Several operations which relate to eToken configuration options require entering either the eToken user password or the eToken administrator password.

Basic eToken Properties

After an eToken is inserted into the USB slot (or if **eToken Properties** is started with an eToken inserted), an icon indicating the eToken is accessible becomes visible in the left panel below the Local Machine icon.



If required for any reason, the information in this window can be copied to the clipboard. Select one or more lines of text and press **Ctrl+C**. To select all the information at once, press **Ctrl+A**. Paste the information to the required application by pressing **Ctrl+V**.

The basic eToken Properties window displays three buttons that enable the user to change a password, rename the eToken and provides the option for a locked eToken to be unblocked (if initialized with an administrator password).

Below these buttons is a table that defines the fields in the basic properties window.

Items marked with (*) apply to the **eToken PRO** and **eToken NG-OTP** only.

Field Name	Field Description
Name	The name given to the token. This name can be changed by clicking Rename eToken...
Type	Product type description.
Model	The eToken model.
Product ID*	USB device product identity.
Firmware version	The version of the eToken firmware.
CARD/OS version*	The eToken smartcard operating system version
User	For the eToken R2 this value is always Present. For the eToken PRO this describes if a User has been defined for this token. Value is either Present or Not Present. A value of Not Present is displayed if the eToken was initialized without defining a user (blank token).
Administrator*	This describes if an Administrator has been defined for this token. Value is either Present or Not Present. A value of Present is displayed if the eToken was initialized with an administrator password
eToken ID	The unique ID for the currently inserted eToken.
Smartcard ID*	The unique smartcard ID for the currently inserted eToken PRO.

Field Name	Field Description
Total memory capacity	The total memory size of the eToken.
Free Memory (R2 ONLY)	The amount of available free memory on the eToken R2.
Largest free block	The size of the largest contiguous block of free memory currently available on the eToken.
Color	This field specifies the color of the eToken. This color is set during the eToken initialization process.
FIPS Mode*	Value can be either On or Off. This field specifies if the eToken was initialized as a FIPS token or not. (Relevant only for eToken PRO models 4.x.5.4)
1024-bit RSA keys count	The number of 1024-bit RSA keys that can be stored on the eToken
2048-bit RSA keys support	If the 2048-bit RSA keys Support Module has been loaded on the eToken, this field will be On , otherwise it will be Off . This field only appears if the CARD/OS version is 4.20 or higher.
HMAC SHA-1 support	If the HMAC SHA-1 Support Module has been loaded on the eToken, this field will be On , otherwise it will be Off . This field only appears if the CARD/OS version is 4.20 or higher.
User login failures limit	The maximum number of consecutive failed log on retries made by the user before the eToken is locked.
Administrator login failures limit	If an administrator password has been initialized, this details the maximum number of consecutive failed log on retries by the administrator before the eToken is locked.
Reader Name	Describes the name of the reader. For USB eTokens, this will always begin with 'AKS ifdh'.

Changing the eToken Password

All eTokens are configured at manufacture with the factory default password. This password is **1234567890**. All eTokens are configured at manufacture with the factory default password. This password is **1234567890**. To ensure strong, two-factor security, and to enable full user functionality, it is important that the user changes the factory default password to an eToken password of the user's own choice, as soon as the new eToken is received. For this reason, the user is forced to change the default password the first time the eToken is used.

After an eToken password has been changed, the new password must be used with the eToken for all eToken applications. It is the user's responsibility to remember the eToken password - without it, the eToken cannot be used for any purpose.

Setting an administrator password on all eTokens enables the administrator to reset the user password if it is forgotten or lost. It is recommended that all eTokens be initialized with an administrator password.



Password Quality

Your password is an important security measure in safeguarding your company's private information. Choosing an effective password is therefore critical.

The best passwords are at least 8 characters long and include upper and lower case letters, punctuation marks and numbers created in a random order. It is not recommended that you use names or birth dates of family members which can easily be discovered.

When changing your password, you can use the eToken Password Quality feature to ensure you are using the most secure password. The eToken Password Quality feature assigns a quality rating to your new password and provides you with tips on how to improve the password. For information on using eToken's Password Quality feature, refer to "How Password Quality Works", on page 100.

➤ **To change the eToken Password:**

- 1 Click **Change password...** on the eToken Properties screen and the following eToken Properties dialog is displayed:



- 2 Enter your current eToken password in the **Current Password** field.
- 3 Enter the new password in the **New Password** field.

NOTE:

As you type the password, the password quality indicator on the right displays how well the new password matches the password quality policy.

If you wish to view more information on why the password quality receives the score shown, click **Show Tips >>**. This expands the window to show a New password tips window. Following these tips will improve the password quality score.

The Password Quality indicator (on the right) provides a percentage score of the quality of the new password. Below the minimum required score, as defined in *etpass.ini*, the password quality indicator remains red. Once the score reaches and passes the minimum required, this color changes to green as displayed:



- 4 Re-enter the new password in the **Confirm Password** field and click **OK**. The eToken password is replaced.

NOTE:

The password quality policy can be enforced if desired. For details on how to enforce such a policy, please refer to "How Password Quality Works", on page 100.

Renaming the eToken

For additional convenience and ease of identification, the eToken name can also be personalized.

➤ **To rename the eToken:**

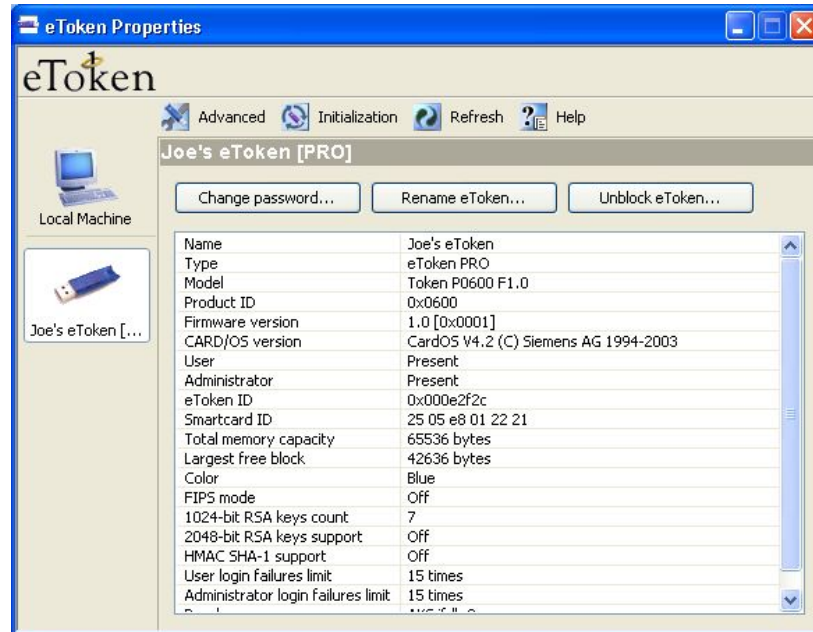
- 1 Click **Rename eToken...** on the eToken Properties screen. Since renaming the eToken requires the eToken password, if this is the first time the eToken password is needed, the following dialog is displayed:



- 2 Enter the eToken password, click **OK** and the **Input eToken Name** dialog is displayed.
- 3 Enter the new eToken name in the **eToken Name** field, as displayed:



- 4 Click **OK** and in the eToken Properties window the new eToken name is displayed:



Unblocking the eToken

Where an eToken has been initialized with an Administrator password, eToken Properties provides the ability to unblock a password on the eToken that may have been locked by attempting to enter an incorrect password too many times.

A challenge response authentication system is used that allows the administrator to unblock the eToken. The user contacts the administrator with the Challenge data from eToken Properties and enters the Response data provided by the administrator. The user then enters a password (either the one previously used or a new one) and the eToken is then unblocked.

➤ To unblock a locked eToken:

- 1 Click **Unblock eToken** in the main eToken Properties window and the following dialog opens:



The image shows the 'eToken Properties' dialog box. It has a blue title bar with the text 'eToken Properties' and a close button. The main area is light beige. At the top, there is a logo for 'eToken'. Below the logo, there are two sections: 'Administrator Login' and 'New Password'. In the 'Administrator Login' section, there is a 'Challenge data' text box containing the hexadecimal string 'DE B0 E7 55 0D 06 E8 41' and an empty 'Response data' text box. In the 'New Password' section, there are two empty text boxes labeled 'Password:' and 'Confirm:'. At the bottom of the dialog, there are 'OK' and 'Cancel' buttons.

- 2 Contact the administrator and provide him with the **Challenge data** (in the example DE B0 E7 55 0D 06 E8 41).
- 3 The administrator provides the **Response data** (in the example 67 D3 AB 06 4E 02 5A 71).
- 4 Enter the **Response data** in the appropriate text box as displayed:



This image shows the same 'eToken Properties' dialog box as before, but now the 'Response data' text box in the 'Administrator Login' section contains the hexadecimal string '67 D3 AB 06 4E 02 5A 71'. All other elements, including the 'Challenge data', the 'New Password' section, and the buttons, remain the same.

- 5 Enter a **New Password** in the **Password** and **Confirm** text boxes as displayed:

The image shows a Windows-style dialog box titled "eToken Properties". It has a blue title bar with a close button (X) in the top right corner. The main area is light beige. At the top, the "eToken" logo is displayed. Below the logo, there are two sections. The first section is titled "Administrator Login" in blue text. It contains two text boxes: "Challenge data:" with the value "DE B0 E7 55 0D 06 E8 41" and "Response data:" with the value "67 D3 AB 06 4E 02 5A 71". The second section is titled "New Password" in blue text. It contains two text boxes: "Password:" and "Confirm:", both filled with eight dots. At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

6 Click **OK** and the eToken is unblocked.

Note:

After providing the Challenge data to the administrator, the user **MUST NOT** undertake any activities that use the eToken until after receiving the Response Data and completing the unblocking procedure.


If any other eToken activity occurs during this process, it will affect the context of the Challenge – Response process and invalidate the procedure.

Administrators can also unblock an eToken by using the Set user password option on the Administrator tab. For details see Setting the eToken User Password on page 81.

Advanced eToken Properties

eToken Properties provides additional functionality that enables setting various advanced configuration options for the eToken user and eToken administrator, if one or both have been defined for the eToken.

For more information on the eToken administrator entity, see Administrator Password, on page 14. For details on initializing the administrator password, see “Initializing using Customizable Parameters”, on page 87.

Click **Advanced**  **Advanced** and for an eToken initialized **without** an Administrator password or an eToken R2 the following dialog is displayed:



For an eToken, initialized with an Administrator password, the following dialog is displayed:



For all eTokens (tokens and smartcards) except the eToken R2, you may log on as a user or as an administrator.

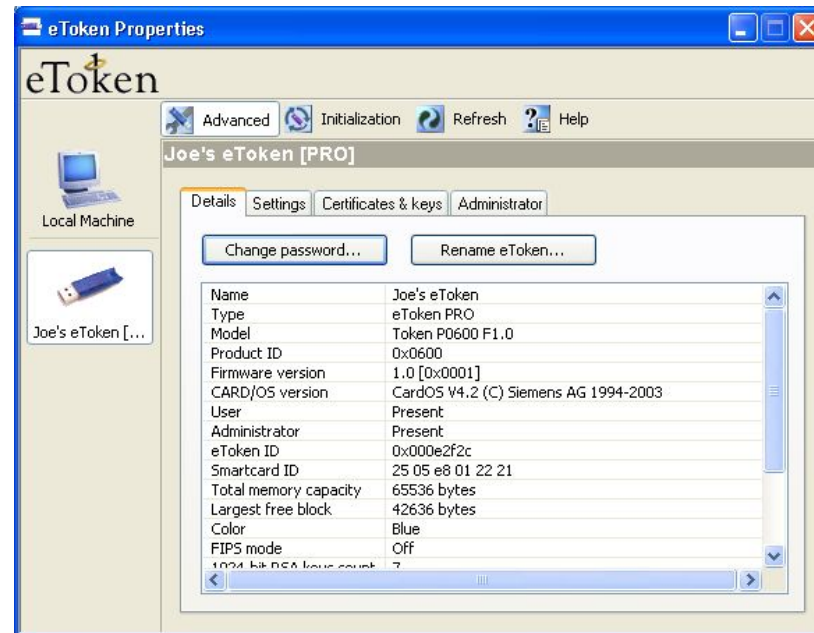
To log in as a user, enter the user password in the **Password** field and click **OK**.

To log in as an administrator, enter the Administrator password in the **Password** field, mark the **Login as Administrator** checkbox and click **OK**.

NOTE:**Administrator login and User functions**

If you log in as an administrator and wish to access functions that require a user password you will be requested to provide the eToken user password. Enter the eToken user password and click **OK**.

The **Advanced Properties** dialog is displayed:



Advanced Properties consists of the following four tabs:

- ◆ Details
- ◆ Settings
- ◆ Certificates and Keys
- ◆ Administrator

NOTE:

Advanced Properties User access

If you log in as a user, you do not automatically have access rights to the Administrator tab. See Administrator tab, on page 78 for details.

Details tab

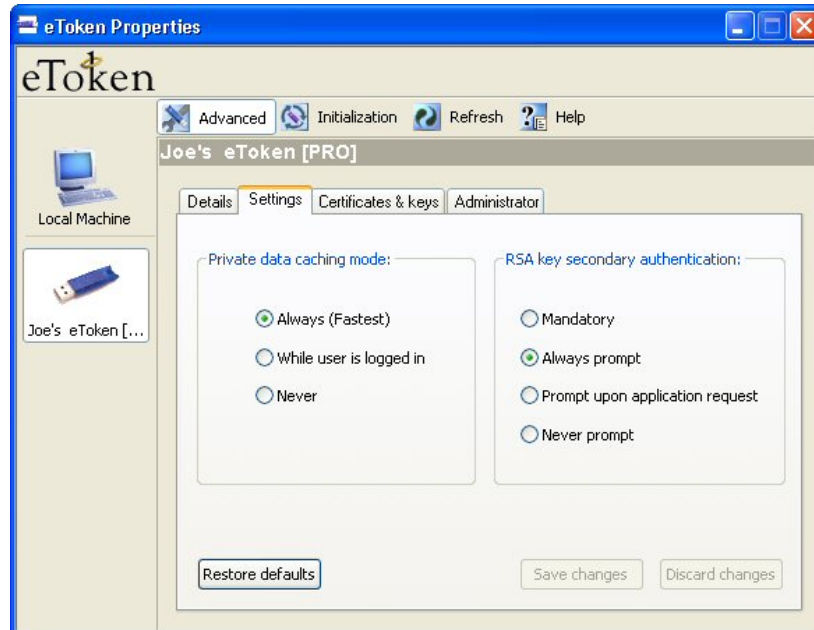
The **Details** tab provides the same information as the **Basic Properties** tab.

See Basic eToken Properties, on page 53, for details.

Settings tab

This tab enables the configuring of settings relating to cache policies and RSA secondary authentication.

Where no administrator entity exists for the token, the user may set these parameters as displayed:



Where an administrator entity exists, the administrator has the ability to allow or disallow the user to modify these parameters. This is done by marking one or both of the **Allow user to modify this option** checkboxes on the Administrator tab (Default - Allow):

◆ Private data caching mode:

In RTE 3.65, public information stored on the eToken is cached by the eToken drivers in order to enhance performance. This group defines the way private information (excluding private keys on the eToken PRO/NG-OTP / Smartcard) can be cached outside the eToken. The following options are available:

- **Always (Fastest)**

Always caches private information in the eToken drivers. This enables fast performance as certain information is cached on the host machine but because of this, this option is less secure than if no cache is allowed.

- **While user is logged in**

Caches private data outside the eToken as long as the user is logged into the eToken. Once the user logs out, all the private data in the cache is erased.

- **Never**

Does not cache private data in the eToken drivers.

♦ **RSA key secondary authentication:**

In RTE 3.65, for the eToken PRO and NG-OTP an option exists to set an additional authentication password for an RSA key. If this option is used, then in addition to having the eToken and knowing the eToken's password, accessing the RSA key requires knowing the password set for that particular key (as displayed below):



This group defines the policy for making use of this secondary authentication of RSA keys. Various options can be set for this policy:

- **Mandatory**

Every time an RSA key is generated, a secondary password for accessing this key is required as displayed:



Clicking **Cancel** will cause key generation to fail. Clicking **OK** generates the key and uses the entered password as the secondary RSA password for that key.

- **Always prompt**

Every time an RSA key is generated, a secondary password for accessing this key is requested as above, however the user can choose to dismiss the prompt (by clicking **Cancel**) and key generation will continue without using a secondary password for the generated RSA key.

- **Prompt upon application request**

This enables applications that wish to use secondary authentication for RSA keys to make use of this feature on the eToken (when creating the key in Crypto API with a user protected flag).

- **Never prompt**

Secondary passwords will not be created for any RSA key and the authentication method will only use the eToken password to access the key.

- ◆ **Restore Defaults**

Clicking **Restore defaults** restores the settings to their default values (private data is always cached and secondary authentication is never allowed). This is only possible if the eToken administrator has defined that the eToken user can modify these parameters.

◆ Save Changes

Saving changes is only possible if the eToken administrator has defined that the eToken user can modify these parameters.

Clicking **Save changes** saves any setting changes that have been made.

◆ Discard Changes

Clicking Discard changes discards any changes made to the private data cache settings or the secondary authentication policy.

Certificates & Keys tab

This tab shows the various certificates, keys and cryptography parameters available on the selected eToken. The following icons are used to identify the various PKI elements:



- Represents a certificate



- Represents an RSA private key



- Represents an RSA private key that will serve as the default (This key is used for Smartcard Logon).



- Represents an RSA private key that requires secondary authentication



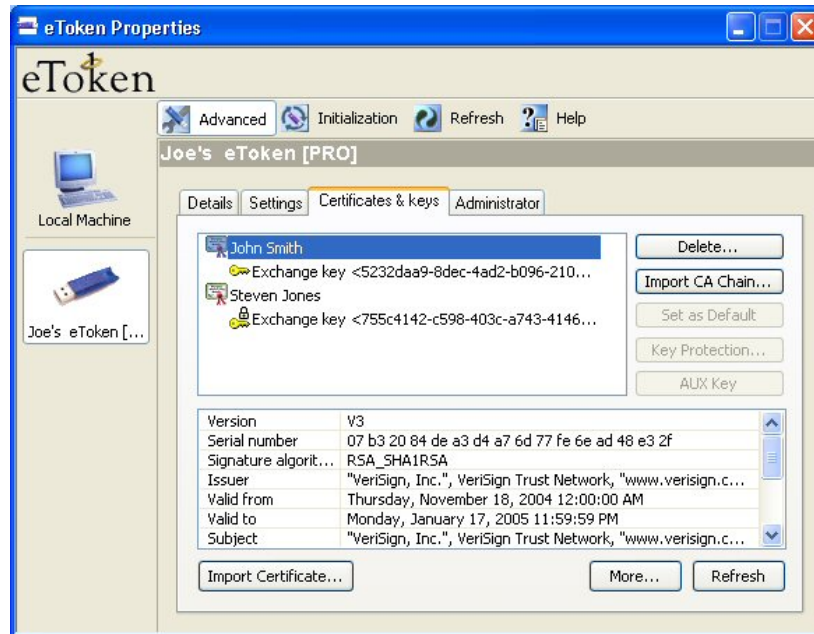
- Represents an RSA private key that requires secondary authentication stored in a default key container



- Represents a CA Certificate that has been imported to the eToken

For more information on secondary authentication for private keys see RSA key secondary authentication, on page 67.

The **Certificate & keys** tab is divided into two windows as displayed:



The top window contains the list of certificates and keys that are stored on the eToken. The list is organized so that if a key corresponds to a certificate, the key appears directly below and to the right of the certificate it relates to.

The bottom window (below the key and certificate list) provides information on a key or certificate selected in the top window. The following tables summarize the available information fields and their meaning for RSA keys and certificates.

Information for RSA Certificates

Field Name	Field Description
Version	The version of the certificate format.
Serial Number	The serial number assigned by the certificate issuer.
Signature Algorithm	The algorithm used for the private key when using it for signing.
Issuer	The name of the organization that issued this certificate.

Field Name	Field Description
Valid from	The date the certificate becomes valid. The certificate cannot be used before this date.
Valid to	The date until which the certificate is valid. The certificate cannot be used after this date.
Subject	A combination of the purpose, conditions and name of the certificate owner might be used as the subject.
Key container	The name of the key container that holds the private key belonging to the certificate's public key.
Key specification	The key specification that defines the purpose of the key.
Public Key	The content of the key that is part of the certificate and is used for encryption.
Certificate Usage	Details for what purposes the certificate is dedicated.
Friendly Name	A combination of the reader name:: and the simple display name of the certificate.


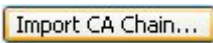


Information for RSA Keys

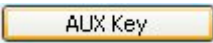
Field Name	Field Description
Algorithm name	Defines the cryptographic algorithm used.
Default KC	The key container used when no specific key container name has been specified when trying to acquire a key container handle.
Key Container	The key container is the place on the token where keys are stored. This field is the name of the key container that holds the selected key.
Key Length	The size of the key in bits.
Key Permissions	Specifies what actions are permitted for this key, e.g. if the key is exportable, the permission would be 0x00000001.

Field Name	Field Description
	eToken PRO keys always have permissions 0x00000000. eToken R2 keys may have permissions 0x00000004.
Secondary Authentication	Details whether the RSA key needs another password in order to be used. Valid for eToken PRO and eToken NG-OTP.
Auxiliary KC	Specifies whether this KC serves as an auxiliary key container (if an auxiliary key container exists.)
Public Key	The public part of the RSA private key that enables encryption of messages, e. g. email, that can be decrypted and read only by the eToken owner (who holds the corresponding private key).

If one of the field names in the top window is selected, all the field information is displayed in the bottom window.

To the right of the certificate and key list window are buttons that perform an action on the currently selected RSA certificate or key as described in the following table:

Button	Description
	Removes the selected RSA key or certificate from the eToken. A confirmation message appears prior to performing this action.
	Imports the complete CA chain of the selected certificate onto the eToken.
	Sets the selected key's key container as the default.
	This key is enabled only when an RSA key created with secondary authentication capability is selected. Key protection... enables setting a new secondary authentication password for the selected key.

Button	Description
	Sets the selected key's key container as the auxiliary.

Note:

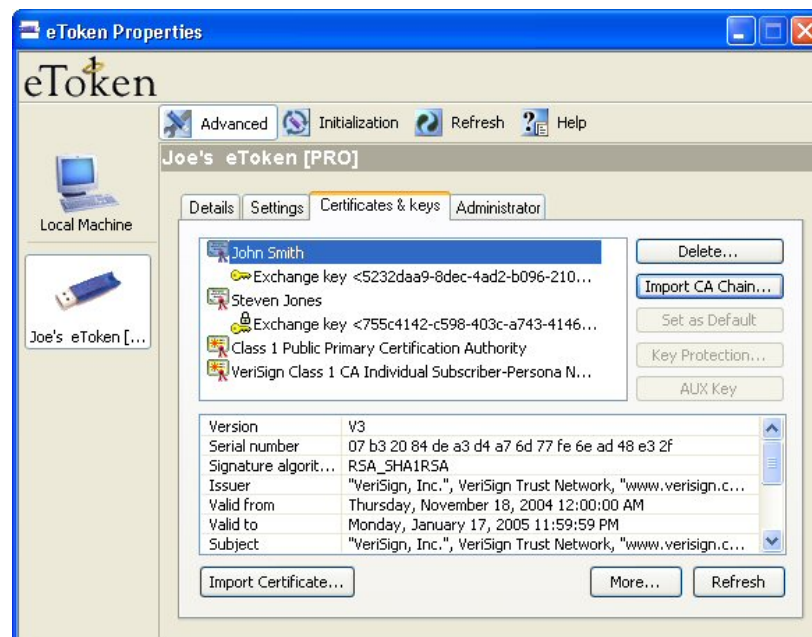
The buttons are only enabled if the action the button executes can be performed on that particular certificate or key.

Import CA Chain

A certificate that is stored on the computer may be part of a hierarchical structure with more than one Certificate in the chain up to the Root CA.

Importing a CA Chain takes the CA certificate and the complete CA Chain up to the root certificate that is stored on the computer and places it on the eToken.

When the **Import CA Chain** button is clicked, the CA Chain is imported onto the eToken and displayed in the certificate and key list window:



A message confirming the import was successful is displayed:



Key Protection

For keys with secondary authentication capability, it is possible to change the secondary authentication password. Click the button and a **Change eToken RSA key password** dialog box opens:



Enter the current and new password in the appropriate text boxes and click **OK**. A confirmation message is displayed:




Aux Key

Certain applications that use CAPI do not say explicitly which key should be used for their operations (e.g. Microsoft VPN). The RTE logic is such that if there is a default key (used for Smartcard Logon) on the eToken, this key will be used for such applications. If no default key exists the RTE arbitrarily chooses a key to use.

Most users do not have multiple keys on their eToken so this mechanism works suitably. However in the case where a user needs to explicitly set a key to be used in such an application, the Aux Key serves this purpose.

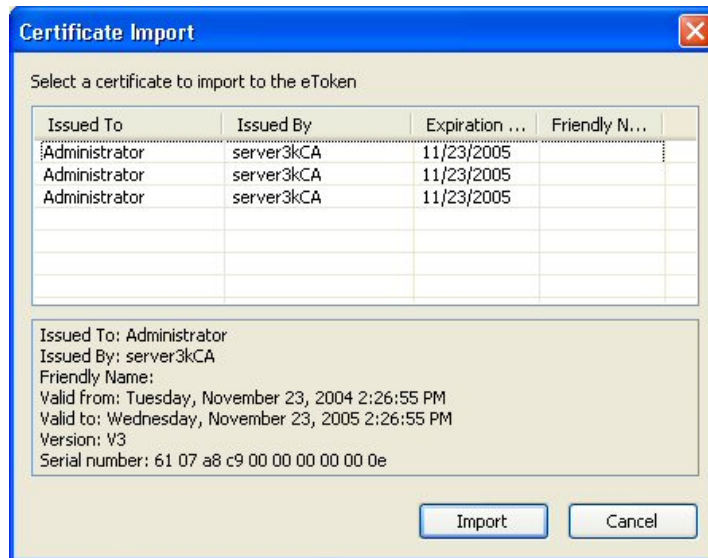
Import Certificate

On the left below the bottom window is the  button. Click this button and the Certificate Import dialog box opens:



Select whether to import the certificate from either your personal store on the computer or a file.

If you select the personal certificate store, a list of available certificates to choose from is displayed:



Not all certificates in the store may be listed. Only certificates that can be imported on to the eToken will be listed. These are:

- ◆ Certificates with a private key already on the eToken
- ◆ Certificates that we might import from the computer together with its private key. (These only work with Windows XP and Windows 2000)

Select which certificate to import and click **Import**. A confirmation message is displayed if the import is successful.

If you want to import from a file, you can import either a PFX or CER file.

If a PFX file is selected (these files are not supported by all Operating Systems), the private key, corresponding certificate and (optionally) CA certificate(s) will be imported to the token. You will be asked to enter the password protecting the PFX file.

In the case of a CER file (which only contains X.509 certificates), the program looks to see if a private key exists on the token. If the private key is found on the token, the certificate is stored with it. If no private key is found, then you are asked if you want to store the certificate as a CA certificate. If you indicate yes, the certificate is stored.

When downloading a certificate to the computer and then importing the certificate to the eToken, the certificate should be removed from the computer and the eToken reinserted before using the certificate to sign and encrypt mail. If this is not done an error occurs while trying to use the certificate on the same computer.

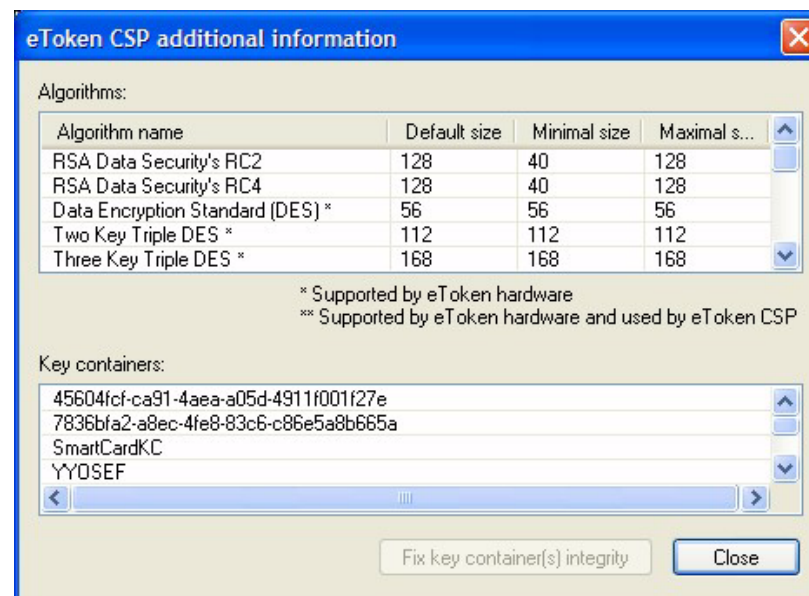
General Buttons

On the right below the bottom window are the **More...** and **Refresh** buttons.



Clicking **Refresh** rescans the eToken for RSA keys and certificates.

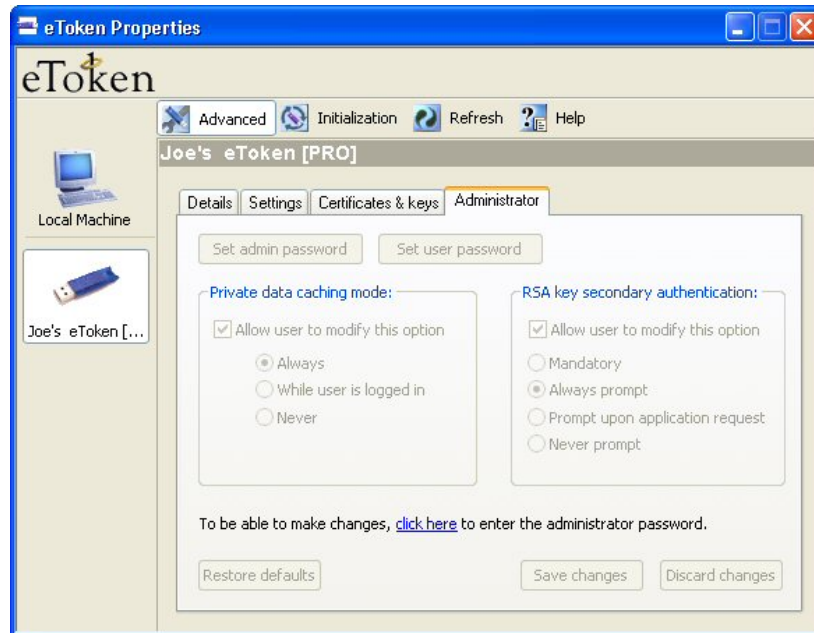
Clicking **More...** opens a pop-up window that provides additional information on the eToken CSP as displayed:



This window contains information on the set of available algorithms and the list of existing key containers.

Administrator tab

If logged on as a user, the administrator tab entries will not be accessible as shown below.

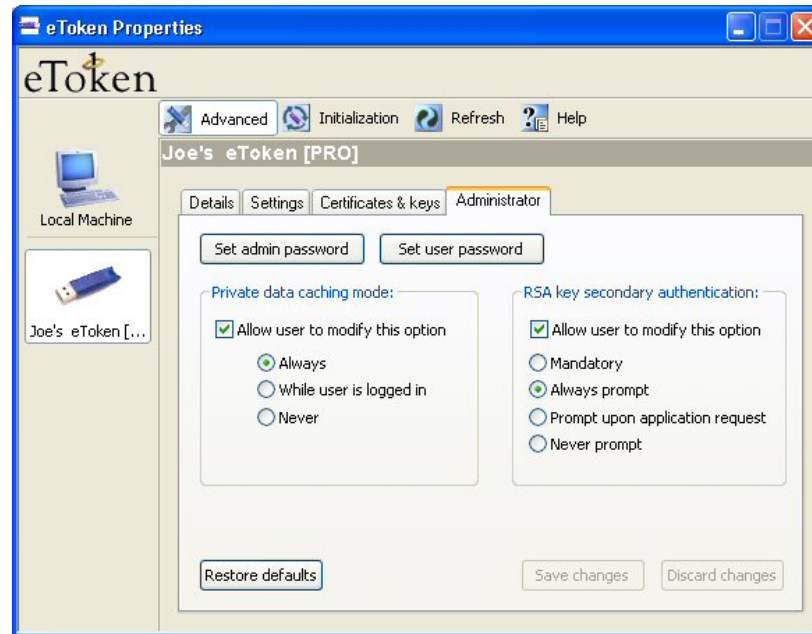


Enabling them requires logging in as the administrator.

Click the link **click here**. This opens a pop-up window requesting the eToken administrator password as displayed:



Enter the correct administrator password and the Administrator dialog is opened and active as displayed:



The Private data caching mode: and RSA key secondary authentication options on the Administrator tab group are exactly the same as on the Settings tab.

However, each group has an additional check box **Allow user to modify this option**. If this option is checked, the eToken user can change these settings on the Settings tab. If the option is unchecked, the user cannot change the settings for these groups. For more information on these options, refer to Settings tab on page 66.

Clicking **Restore Defaults** will restore the default configuration for the Administrator settings (private data is always cached and user can modify private data cache policy, secondary authentication for RSA keys is set to “never prompt” and user can modify RSA keys secondary authentication policy).

Clicking **Save Changes** will save all changes made to the Administrator settings.

Clicking **Discard Changes** will discard any changes made to the administrator settings.

If logged on as an eToken administrator, you have the option to set the eToken user password and change the eToken administrator password.

Setting the eToken Administrator Password

Click **Set admin password** to change the eToken administrator password and the following dialog is displayed:

A screenshot of a Windows-style dialog box titled "Change eToken administrator pass...". The dialog has a blue title bar with a close button (X) in the top right corner. The main area has a light beige background with the "eToken" logo at the top. Below the logo are three text input fields labeled "Current Password:", "New Password:", and "Confirm Password:". At the bottom of the dialog are two buttons: "OK" and "Cancel".

Enter the current eToken administrator password in the **Current Password** field.

Enter the new eToken administrator password in the **New Password** field.

Reenter the new administrator password in the **Confirm Password** field and click **OK**.

The eToken administrator password is replaced with the new administrator password.

Setting the eToken User Password

When logged in as an eToken administrator, you have the option to reset the eToken user password and error retry counter. This is usually used in cases where the eToken user password has been forgotten or locked.

Click **Set user password** to change the current user password. The following dialog is displayed:

The image shows a Windows-style dialog box titled "Set eToken Password". It has a blue title bar with a close button (X) in the top right corner. The main area has a light beige background with the "eToken" logo at the top. Below the logo, there are three input fields: "New Password:" followed by a text box, "Confirm Password:" followed by a text box, and "Set error retry counter:" followed by a spin box currently showing the number "5". At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

Enter the new user password for the eToken in the **New Password** field.

Re-enter the new user password in the **Confirm Password** field

The error retry counter is set to 5 in this dialog box because that is what is set on the sample eToken although the default is 15 retries. Select a number between 1 and 15 for the number of retries you want on this eToken.

The eToken user password is changed to the newly entered password and the retry counter is set accordingly.

Chapter 7

eToken Initialization

The eToken Initialization feature removes all files stored on an eToken PRO, eToken NG-OTP or eToken Smartcard since manufacture, frees up available memory, and resets the default eToken password, allowing administrators to initialize the eToken according to specific organizational requirements or security modes.

Initializing an eToken is useful, for example, after an employee has left a company. It completely removes the employee's individual eToken password, certificates and other personal data from the eToken, leaving it ready to be set up and used by another employee.

The eToken Initialization tool is also useful for software developers, since it can be used during testing to reinitialize an eToken PRO/NG-OTP (CardOS/M4), and to specify various settings.

About This Chapter

This chapter includes the following sections:

- ◆ “Initializing the eToken”, on page 84, details how the initialization process works.
- ◆ “Initializing using Customizable Parameters”, on page 87, explains all the custom parameters that can be configured and how to use them for initialization.

Initializing the eToken

The eToken Initialization feature restores an eToken PRO/NG-OTP or Smartcard to its initial state. It also enables the initialization of multiple eTokens and sets the initial eToken configuration, including:

- ◆ User password
- ◆ Administrator password (optional)
- ◆ Retry counters (for user and administrator passwords)
- ◆ Initialization key
- ◆ FIPS or non-FIPS mode

NOTE:

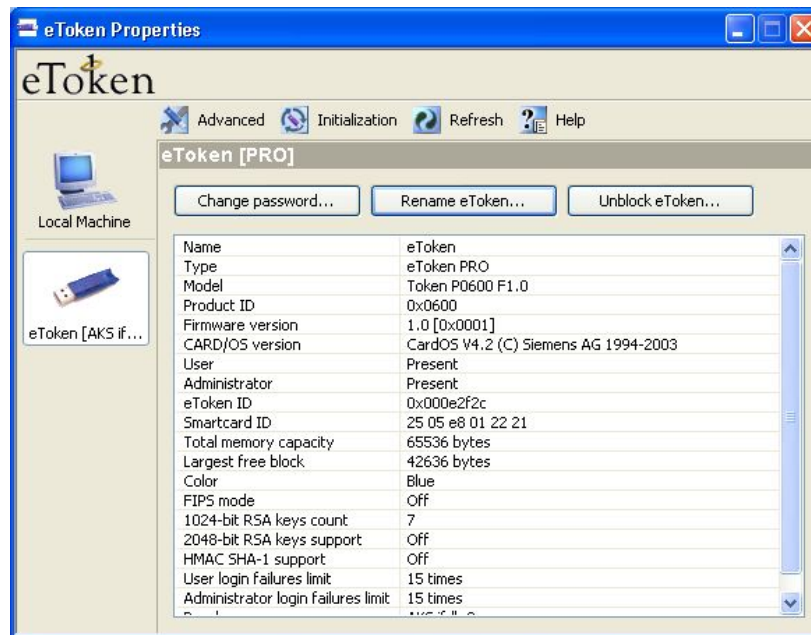
The eToken Initialization feature is designed for use with the eToken PRO, eToken NG-OTP and smartcards.

The initialization process loads the Aladdin file system and files on the eToken. The eToken PRO (version 4.x.5.4) can also be configured in FIPS mode.

FIPS stands for Federal Information Processing Standards and is a US government approved set of standards designed to improve the utilization and management of computer and related telecommunication systems. After initializing the eToken PRO in this mode, it will be FIPS compliant.

➤ **To initialize an eToken:**

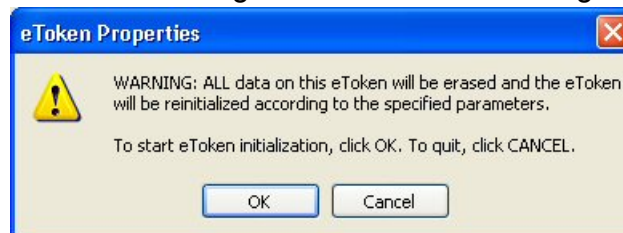
- 1 Open eToken Properties (from the *Start menu*, select *Programs\eToken\eToken Properties*) and insert an eToken (if not already inserted). The eToken Properties screen is displayed:



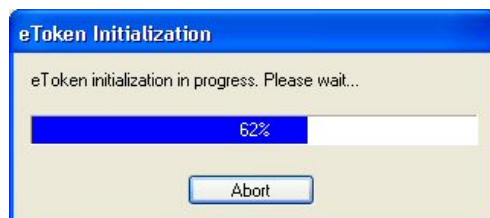
- 2 Click **Initialize** and the eToken Initialization dialog box opens:



- 3 Click **Start** to begin initialization. A warning is displayed:



- 4 Click **OK** to start the initialization process. During initialization a progress bar is displayed:



- 5 When initialization is completed, a confirmation message is displayed:

**Note:**

The Initialization process resets the password to the default unless this is changed in the Set Parameters dialog.

Initializing using Customizable Parameters

Using customizable parameters, you can select specific parameters that will apply to certain eTokens. These parameters may be necessary if you wish to use the eToken for specific applications or if you require a specific user or administrator password on all the eTokens in the organization.

➤ **To initialize an eToken using customizable parameters:**

- 1 Click **Parameters...** on the **eToken Initialization** dialog box and the **eToken Initialization Parameters** dialog box opens:

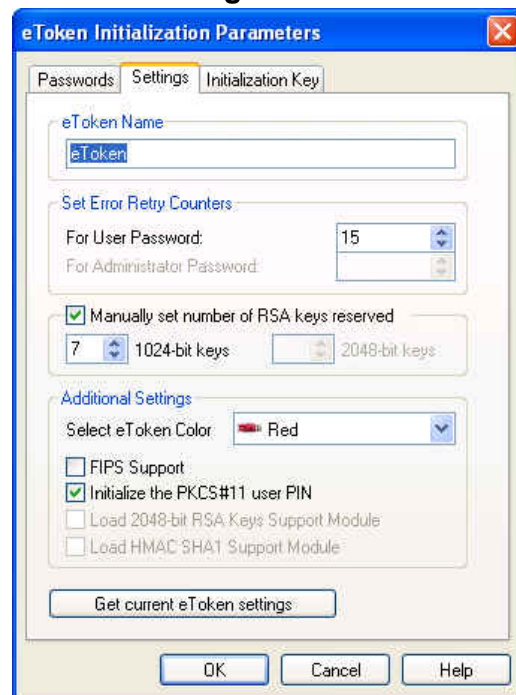
The screenshot shows the 'eToken Initialization Parameters' dialog box. It has three tabs: 'Passwords', 'Settings', and 'Initialization Key'. The 'Passwords' tab is selected. Inside the dialog, there are three main sections. The first section, 'Initialize eToken', has a checked checkbox. The second section, 'Create user password', has two text input fields labeled 'Enter password' and 'Confirm'. The third section, 'Administrator password', has an unchecked checkbox and two text input fields labeled 'Enter password' and 'Confirm'. The fourth section, 'Current Password', has a text input field and a checkbox labeled 'Use Password'. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

- 2 Ensure that the **Initialize eToken** check box is enabled. This enables the **Create user password** and **Administrator password** sections.

If the **Initialize eToken** check box is **not** selected, the eToken will be initialized as a blank token. Only the support modules and Initialization key can be set.

To initialize the eToken to work with eToken software you must select **Initialize eToken**. This allows all initialization parameters to be set.

- 3 Set the new eToken User Password in the **Enter password** field (the default eToken Password is **1234567890**) and reenter it in the **Confirm** field.
- 4 If you wish, select **Create administrator password** in order to set an **Administrator Password** for the eToken (minimum password length must be 4 characters).
- 5 Set the Administrator Password in the **Enter password** field and re-enter it in the **Confirm** field.
- 6 Select **Use Password** and enter the current eToken password in the text box. If the check box is not selected the password used for logging into the eToken can be used. The **Current Password** option is relevant only for eTokens previously initialized in FIPS mode. This password must be either the current user password or current administrator password.
- 7 Click the **Settings** tab and the following dialog is displayed:



- 8 Enter a name for the eToken in the **eToken Name** field. (If no name is entered, the default name will remain **eToken**).
- 9 Two error retry counters exist. The **User Password** retry counter is always enabled while the **Administrator Password** retry counter is enabled only if an Administrator password was set on the previous dialog. To **Set Error Retry Counters**, select either or both **For User Password** or **For Administrator Password** (if enabled) and enter a value between 1 and 15 in the scroll box.

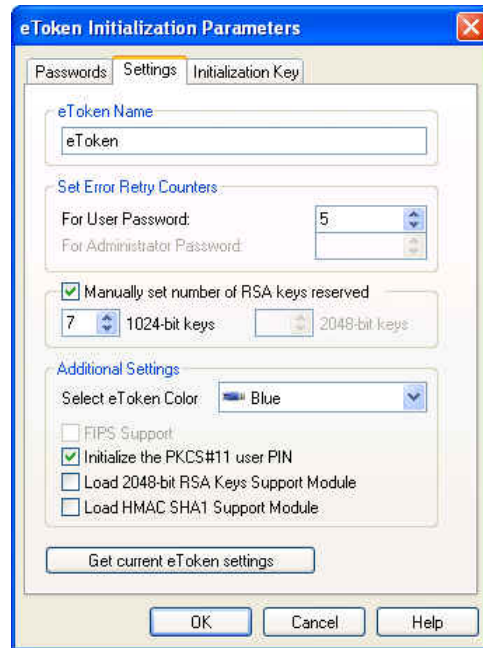
The retry counter specifies the number of times the user / administrator can attempt to log in to the eToken with an incorrect password before the eToken is locked.

When the retry counter is not specified, the default setting for the maximum number of incorrect login attempts is **15**.

- 10 If desired, select **Manually set number of RSA keys reserved** and enter a value in the appropriate scroll box.

The RSA Keys Reserved number specifies the maximum number of 1024-bit and/or 2048-bit RSA keys that can be stored on the eToken. The remaining memory on the eToken is used for all other applications and not for storing additional 1024-bit and/or 2048-bit RSA keys.

If the 2048-bit RSA Keys Support Module was loaded, the number of these keys that are reserved can be seen when you manually set the number of 1024-bit RSA keys as displayed below:



In older eTokens, the “color” of the eToken referred to a software definition stored on the smartcard and not necessarily the true eToken color. Therefore it was possible to use the **Select eToken Color option** to arbitrarily select a different “color” for the eToken.

In newer eTokens, the color reference is burned during production and **cannot** be changed. It is always the real color of the eToken.

- 11 If enabled, select a new color from the **Select eToken Color** drop down box.
- 12 Clear **Initialize PKCS#11 user PIN** if you do not want the eToken PKCS#11 library to indicate that the eToken PIN is initialized. In this case, some applications that use the PKCS#11 API cannot work with the eToken until its user PIN is initialized.

Note:

If the eToken is not PKCS#11 initialized, you can change the user password and this will set the PKCS#11 initialization without having to reinitialize the eToken.

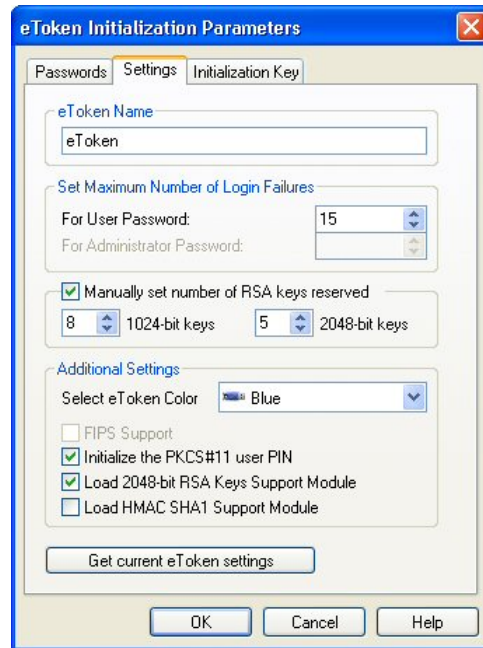
13 Select **Load 2048-bit RSA Keys Support Module** or **Load HMAC SHA1 Support Module** if you need to use either of these modules:

- **2048-bit RSA Keys Support Module** - Allows the user to work with 2048-bit keys (for eToken with CardOS 4.20 and higher only).
- **HMAC SHA1 Support Module** - Allows the user to use the HMAC SHA1 algorithm for OTP tokens (for eToken with CardOS 4.20 and higher only).

For eTokens with CardOS 4.20, the **2048-bit RSA Keys Support Module** had to be specifically enabled during initialization. Due to memory restrictions on the card, this option and the **HMAC SHA1 Support Module** are mutually exclusive.

The newer CardOS 4.20B and 4.30B versions have built-in support for RSA 2048-bit keys and the support module is automatically loaded during initialization (irrespective of whether the option is selected or not). Having this support does NOT exclude support for HMAC SHA-1 and this can be enabled by selecting the **Load HMAC SHA1 Support Module** when setting initialization parameters.

14 Click **Get current eToken settings** to automatically fill the settings of the currently connected eToken.



- 15 Select **OK** to close the **eToken Initialization Parameters** dialog box.
- 16 Click **Start** to begin the initialization process.

Setting or Changing the Initialization Key

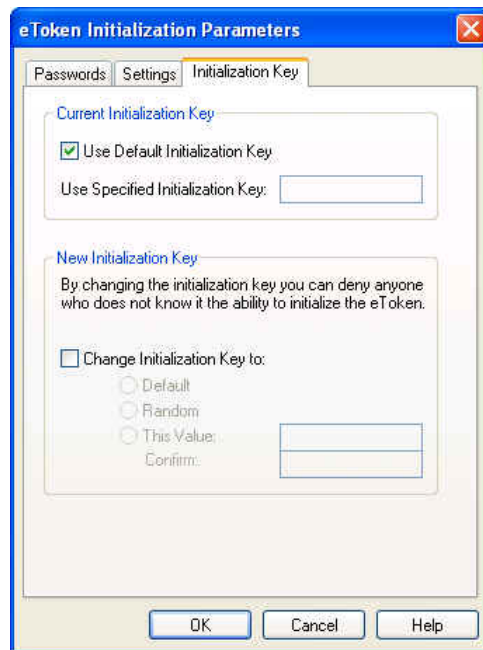
An Initialization Key protects the initialization process and without knowing the current initialization key of the eToken, no initializing of that specific eToken can occur.

If the Initialization Key of the eToken you wish to initialize is different from the default Initialization Key you must set the current Initialization Key.

By changing the Initialization key, you can control and secure the initialization process. When the default Initialization key is not employed, only the person who knows the Initialization key is authorized to initialize the eToken.

➤ To set or change the Initialization Key parameter:

- 1 Click the **Initialization Key** tab and the following dialog is displayed:



- 2 If the current initialization key is not the default initialization key, clear the **Use Default initialization Key** check box and enter the current initialization key in the **Use Specified initialization Key** text box.
- 3 To create a new initialization key, in the **New Initialization Key** section select the **Change Initialization Key to:** check box.

Select the **Default** option to change the initialization key back to the default initialization key.

or

Select the **Random** option to use a randomly created initialization key.

NOTE:

Selecting this option prevents future reinitializing of the eToken.

or

Select the **This Value** option to use a new initialization key. Enter the new initialization key and re-enter it in the **Confirm** field to confirm the new initialization key.

- 4 Select **OK** to close the eToken Initialization Parameters dialog box.
- 5 Click **Start** to begin the initialization process.

Multi-token Initialization

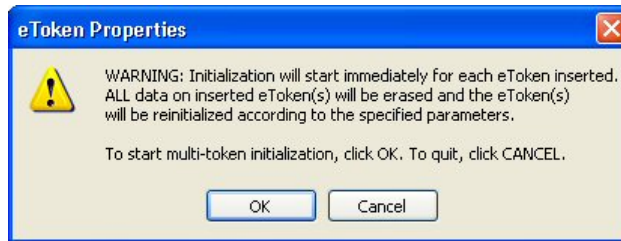
When initializing eTokens, it is possible to initialize one only or to initialize multiple eTokens one after another using the same parameters for all the eTokens. If the **Multi-token Initialization** option is selected, all eTokens plugged into the computer are initialized. Each new eToken inserted is automatically initialized. If a hub that accommodates multiple eTokens is used, all the eTokens inserted in the hub at any given time will be initialized. The number of eTokens that can be simultaneously initialized depends on the number of virtual readers that have been installed, See Readers Management on page 50.

➤ **To automatically initialize each eToken inserted:**

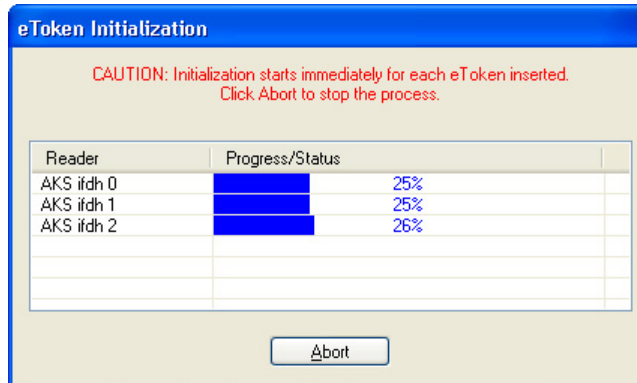
- 1 Plug-in the eToken or eTokens you want to initialize and click **Multi-token Initialization** on the main eToken Properties screen. The **Multi-token Initialization** dialog box opens:



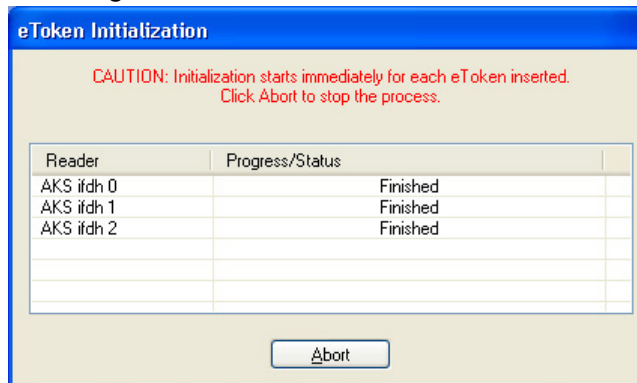
- 2 Set the various parameters as detailed in the previous section and click **OK** to close the eToken Initialization parameters. These parameters will apply to ALL the eTokens initialized during this session.
- 3 Click **Start** to start the **Multi-token Initialization** procedure. A warning is displayed:



- 4 Click OK to start multi-token initialization. A progress bar for each eToken being initialized is displayed during the process:



- 5 When the initialization of the inserted eToken is completed, the Progress/Status field states **Finished** as displayed:



- 6 Click **Abort** to end the **Multi-token Initialization** process.

Warning:

When the Automatic eToken Initialization is activated, Initialization starts immediately for every eToken inserted. Only when clicking **Abort** will the Automatic Initialization stop.

Stopping the Initialization Process

As soon as the initialization process starts, the eToken becomes unusable until it is successfully initialized.

The **Abort** button, which is enabled only after the initialization has started, lets you halt the process and revise the settings for the eToken before reinitializing it. **This is not recommended as it may lead to the eToken becoming corrupted and unable to be initialized in the future.**

Stopping the initialization leaves the eToken in an undefined state. You *must* initialize the eToken in order for it to be usable.

Password Reset

By default, initializing an eToken resets its password to the default - **1234567890**. To ensure maximum security, each eToken should be assigned its own individual password before being used in a live environment.

The eToken Properties configuration tool lets you change the eToken password as required. See eToken Properties on page 45 for details.

NOTE:

All eToken initialization parameters EXCEPT passwords and keys are stored on the user's computer and the user's choices are remembered when the computer is restarted.

Troubleshooting

The following Error Messages may appear when trying to initialize an eToken.

Message	Reason
The selected format type cannot be applied to the eToken R2	Only eToken PRO and eToken NG-OTP tokens can be initialized using this feature.
This eToken does not support FIPS mode	The eToken version used is not FIPS compliant. Use an eToken with firmware version 4.1.5.4 or 4.2.5.4 ONLY.
This eToken does not support 2048-bit keys	Only eTokens with CardOS M4.20 or later support this feature.
This eToken does not support the HMAC-SHA1 calculation	Only eTokens with CardOS M4.20 or later support this feature.
The HMAC_SHA1 and RSA-2048 features cannot be supported together	For eTokens with CardOS 4.20 only one feature can be loaded on these eToken models - not both.
The current password is not valid	Initialization can only be performed with a valid current password. Input a valid current password. (eToken was probably in FIPS mode).
Cannot initialize this eToken without the current password	User must enter the current password to initialize the eToken. (eToken was probably in FIPS mode).
The initialization key is not valid	Input a valid initialization key to initialize eToken.

The following general troubleshooting procedures may be required.

Problem	Answer
I have a FIPS token and try to initialize it, but the initialization fails.	When initializing a FIPS token, you need to set the Use Current Password field (with the current user or administrator password) regardless of the newly chosen format type.
I set the eToken Password value (for a user or administrator). I then close the application. The next time I open eToken Properties, the password I had set previously is no longer used.	eToken Properties saves certain set parameters on the PC. However any security-related information (such as passwords) is not saved.

Chapter 8

eToken Password Quality

The eToken Password Quality feature enables the creation of a password quality policy for the eToken. The Password Quality feature uses definitions from the ETPass.ini file, located in the system folder, to define the quality of a given password. The Password is rated based on different parameters that have been assigned penalties, which are used to calculate the password rating.

Note:

The ETPass.ini file can be edited and distributed over the network to all user stations using a logon script.

About This Chapter

This chapter includes the following sections:

- ◆ “How Password Quality Works”, on page 100, explains the functioning of the password quality feature.
- ◆ “Password Quality Parameters”, on page 104, details all the parameters used in this feature.
- ◆ “Editing the Password Quality Configuration File”, on page 108, provides a step-by-step explanation of how to edit the configuration file to your own specifications
- ◆ “Using Your Own Configuration File”, on page 111, explains how to create and use your own configuration file..

How Password Quality Works

The Password Quality feature allows the administrator to edit the password quality parameter set and save it in the eTPass.ini file. While editing, the administrator has the facility to check any proposed password's quality value.

Once the quality parameters have been set, any future passwords are automatically checked against these parameters to determine the password's level of acceptability.

The current password's quality value (as shown on the progress bar) can vary from 0% - an "absolutely bad" password - to 100% - an "absolutely good" password. This is shown as a percentage in the progress bar. Once the **MinimalQuality** parameter is achieved, this progress bar changes from red to green and means the password can be used.

Calculating the Password Quality Value

The Password Quality calculation algorithm uses a complex set of parameters to determine the exact quality of the input password and whether or not the eToken password meets the minimum criteria for acceptability. The password quality value depends on the length and properties of the input password. The exact algorithm used is dependent on the RTE version and may change in the future.

In determining the final password quality, a three-stage calculation is performed:

- ◆ A password minimal length check is firstly undertaken.
- ◆ The base password quality is then determined using the optimal password length.
- ◆ Finally the penalty parameters are invoked to determine whether any additional penalties on the chosen password will be applied.

Password Dictionary File

The file is available to be included in the password quality system to resist a “dictionary attack” that may be attempted against the eToken password. The administrator can include frequently used (and therefore weak) passwords, to which the **Dictionary** parameter refers when examining the proposed password. Use of one of these words results in a **CheckDictionary** penalty. and use of a word similar to one of the dictionary words results in a **LikeDictionary** penalty.

Change Password Policy

DefaultPassChange, **Expiry**, **ExpiryEnforce** and **MinChangePeriod** parameters impact the change password policy. When a user logs in, the system checks if the password is 1234567890 (default). If so, the value of the **DefaultPassChange** parameter defines whether this action can be successfully performed. This parameter can be:

- ◆ **'none'** - allows this password.
- ◆ **'warning'** - shows a warning and suggests to change it (this change is not mandatory)
- ◆ **'enforce'** - login with 1234567890 is not allowed. To log in the user must change this password before login.

The parameter **Expiry** defines the number of days after which a warning about the password expiration will be shown.

The parameter **ExpiryEnforce** defines the number of days after which the current password must be changed. Failing to change the password will result in login failure.

The parameter **MinChangePeriod** defines the number of days before which changing the eToken password is prohibited.

Installed values of PQ parameters:

MinimalLength	4
WarningLength	6

OptimalLength	12	
SmallPassword	-5%	
Duplicates	-20%	
Repeating	-20%	
NoLowerCase	-5%	
NoUpperCase	-5%	
NoPunctuation	-5%	
NoDigits	-5%	
DigitsOnly	-5%	
PhonesAndSerialNumbers	-5%	
WhiteSpaces	-100%	(denied)
NonPrintable	-100%	(denied)
KeyboardProximity	-10%	
KeyboardProximityBase	3	
ABCOrder	-10%	
ABCOrderBase	3	
Dictionary	- not defined	
CheckDictionary	-100%	(denied)
LikeDictionary	-80%	
Expiry	360	

ExpiryEnforce	0	(never)
MinChangePeriod	0	(never)
SaveOldPasses	3	
CheckOldPasses	0%	(does not check)
CheckCurrPass	-100%	(denied)
DefaultPassChange	enforce	
MinimalQuality	30%	

Password Quality Parameters

The various parameters all have a variable assigned value determined by the Administrator in accordance with the organization's password quality policy. The value of the parameter will be determined in accordance with the importance that each parameter has in the determination of that policy.

A brief description and explanation of each parameter listed in the ETPass.ini file follows:

◆ **ABCOrder** **Consecutive letters penalty**

Use of consecutive letters in the new password.

◆ **ABCOrderBase** **Consecutive letters base**

The minimum number of consecutive letters in the new password that will create a penalty.

◆ **CheckCurrPass** **Current password penalty**

Use of a the current password as the new password.

◆ **CheckDictionary** **Dictionary password penalty**

Use of a word from the dictionary file as the new password.

◆ **CheckOldPasses** **Old password check**

Use of a password that was previously used as the new password.

◆ **DefaultPassChange** **Change password policy**

The policy for using the eToken default password at login time allows for 3 modes:

- **None** - No action if default kept.
- **Warning** - A warning message will be displayed.
- **Enforce** - User cannot use default and must change it.

◆ **Dictionary** **Dictionary file name**

The name of the dictionary file which contains a list of poor passwords as determined by you.

Each line in the file represents one password. Different passwords need to be on different lines.

See CheckDictionary for penalty details.

◆ **DigitsOnly** **Digits only penalty**

Use of digits only as the new password.

◆ **Duplicates** **Duplicates penalty**

Use of duplicate characters in the new password.

◆ **Expiry** **Password expiry warning**

The specified number of days to use the password before it is recommended to be changed.

◆ **ExpiryEnforce** **Password expiry (days)**

The current password is valid only for the specified number of days and will expire at the end of this period.

◆ **KeyboardProximity** **Keyboard proximity penalty**

Use of keyboard letters next to each other in the new password.

◆ **KeyboardProximityBase** **Keyboard proximity base**

The minimum number of letters next to each other on the keyboard in the new password that will create a penalty.

◆ **LikeDictionary** **Dictionary like password penalty**

Use of a word that is like a word from the dictionary file (e.g. one character different from a dictionary word) as the new password.

◆ **MinChangePeriod** **Minimum change period**

The minimum number of days **before** the password can be changed.

◆ **MinimalLength** **Minimal password length**

The minimal password character length is as specified. Less than this length is not allowed.

◆ **MinimalQuality** **Minimal password quality**

The minimum percent password quality that is allowed.

◆ **NoDigits No digits penalty**

Use of no digits in the new password.

◆ **NoLowerCase No lower case penalty**

Use of no lower case characters in the new password.

◆ **NonPrintable Invalid symbols penalty**

Use of non-printable symbols in the new password.

◆ **NoPunctuation No punctuation penalty**

Use of no punctuation marks in the new password.

◆ **NoUpperCaseNo upper case penalty**

Use of no upper case characters in the new password.

◆ **OptimalLength Optimal password length**

The optimal password character length is as specified. Use of less than optimal character password length determines the basis of the penalty calculation (e.g. using an 8 character password in a 10 character optimal length password will result in an 80% quality level, before other penalties).

◆ **PhonesandSerialNumbers Phones and serial numbers penalty**

Use of telephone, social security, serial, license numbers etc. in the new password.

◆ **Repeating Repeating penalty**

Use of repeating characters in the new password.

◆ **SaveOldPasses Save old passwords**

The number of previously used passwords that will be encoded and stored on the eToken and not available for reuse.

◆ **SmallPassword Small password penalty**

The penalty in the quality score for each character below the length of WarningLength.

◆ **WarningLength** **Warning password length**

The length of the password below which a warning is issued in the quality check.

◆ **WhiteSpaces** **White spaces penalty**

Use of white spaces instead of characters in the new password.

Editing the Password Quality Configuration File

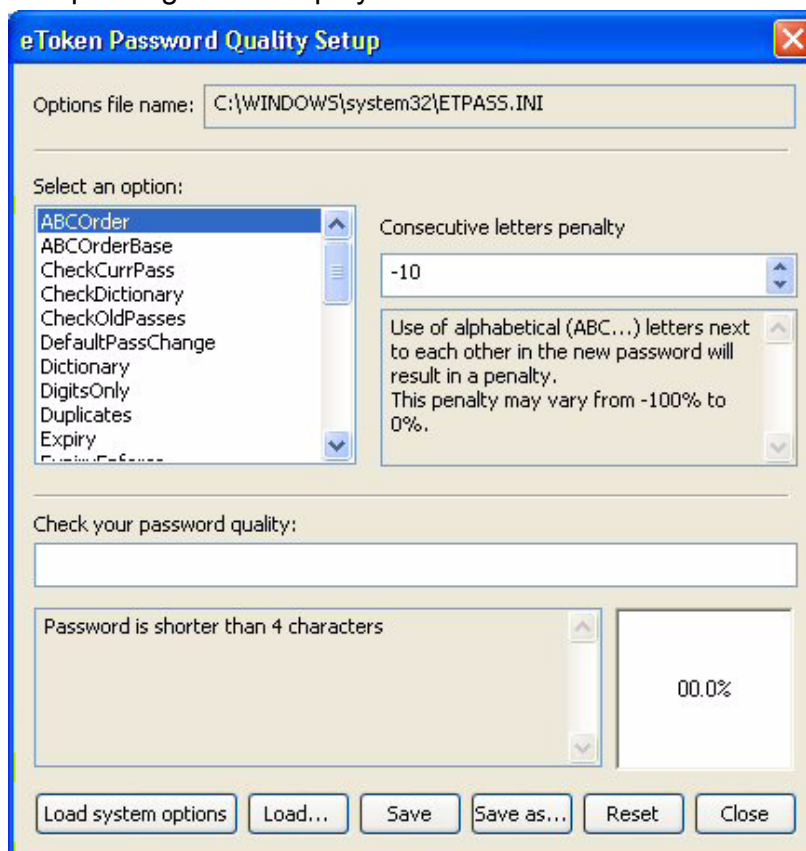
The section describes how to edit the default password quality configuration file.

➤ **To edit the password quality configuration file:**

- 1 Open eToken Properties (from the *Start* menu, select *Programs\leToken\leToken Properties*). The eToken Properties main screen is displayed:

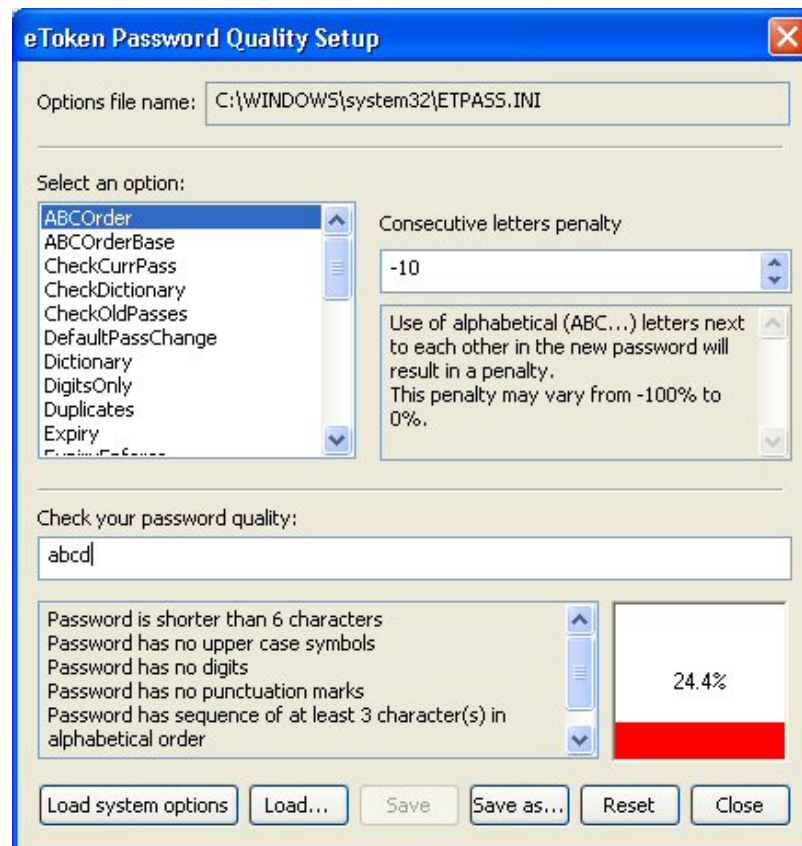


- 2 Click **Password Quality** and the eToken Password Quality Setup dialog box is displayed:



- 3 Click the **Load systems options** button to load the default password quality configuration file.
- 4 Select the password quality parameter you would like to edit from the list on the left side. For example, the LikeDictionary parameter assigns a penalty of -80 to passwords that are similar to words found in a dictionary.
- 5 Modify the penalty assigned for the selected parameter by entering values directly into the field to the right of the parameters list, or by using the accompanying arrow buttons.

- 6 Enter your password in the **Check your password quality** field to view the effect of the parameter on your password. Manipulating the penalty changes the password rating displayed. Elements of password quality that are not being fulfilled are displayed at the bottom of the dialog, along with a percentage rating of your password, as shown in the example below:

**Note:**

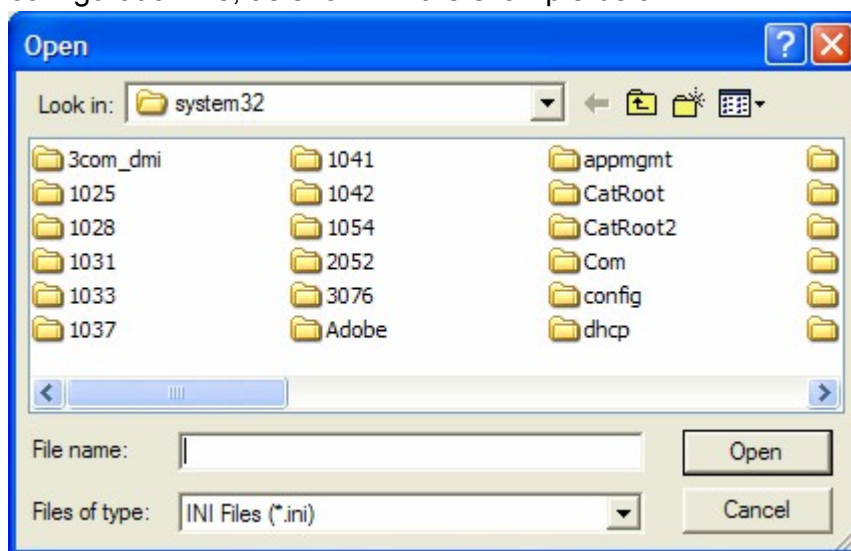
If you are not satisfied with the changes you have made to the password quality configuration file, click the **Reset** button to revert to the default settings.

- 7 When you are finished editing the password quality configuration file, click the **Save** button to save your changes to the current system file. Click the **Save as...** button to save the password quality configuration file to a different location.

Using Your Own Configuration File

➤ **To use your own password quality configuration file:**

- 1 Open eToken Properties and click **Password Quality**. The eToken Password Quality Setup dialog box opens.
- 2 Click the **Load** button to load the default password quality configuration file or to browse for a different password quality configuration file, as shown in the example below:



- 3 Edit the selected password quality configuration file, as described in “Editing the Password Quality Configuration File”, on page 108.
- 4 When you are finished editing the password quality configuration file, click the **Save** button to save your changes.
- 5 Click the **Save as...** button to save the password quality configuration file to a different location.

Chapter 9

Troubleshooting

This chapter offers advice and proposes solutions to problems that you may encounter when installing or using eToken.

About This Chapter

This chapter includes the following sections:

- ◆ “Problems and Possible Solutions”, on page 114, lists the problems that might arise, and suggests their causes and solutions.
- ◆ “Checking USB Support”, on page 116, explains how to check whether USB support is enabled in the BIOS for your system.
- ◆ “Technical Support”, on page 118, provides contact information for technical assistance.

Problems and Possible Solutions

The following table lists the possible causes of each problem, and suggests the appropriate solutions.

Table 1: Problems, Diagnoses and Solutions

	Problem	Possible Diagnosis	Solution
1	Operating system identifies new hardware, but fails to recognize it as a USB device.	The eToken was inserted into the USB port before installation was finished.	Remove the eToken from the port and reinsert.
		Installation was not successful, or the driver was not installed correctly.	Remove the eToken RTE installation, if necessary, and reinstall.
2	LED on eToken does not light up.	The USB is not enabled in the BIOS. See Checking USB Support, on page 116, for details.	Enable the USB in the BIOS. If necessary, consult your technical support services supplier.
		The eToken was inserted during installation.	Remove the eToken and reinsert it in the USB port.
		The eToken is defective.	Obtain a new eToken. Contact your local Aladdin office.
3	Application does not recognize the eToken.	Errors in the application.	Check the application for errors.
		The eToken is defective.	Obtain a new eToken.

	Problem	Possible Diagnosis	Solution
4	Operating system displays the “New Hardware” message when a different USB port is used.	Windows automatically recognizes a new port when it is used for the first time, including ports connected via a hub.	This is normal operating system behavior and needs no further action. The current eToken installation is valid for all USB ports.
5	RTE installation failure on Windows NT 4.0.	The USB port is not enabled in the BIOS.	Make sure the USB interrupt is enabled in the BIOS settings.

Checking USB Support

Most Operating Systems include automatic USB support.

Certain early versions of Windows 95 do not come with USB support. and in these cases, the Windows 95 OSR2 service pack must be installed to provide the necessary USB support.

Windows NT 4.0 also does not come with automatic USB support. In order for the system to recognize the USB port and your eToken, USB support must be enabled in the BIOS. Your technical support services supplier may need to make the necessary changes to your system setup.

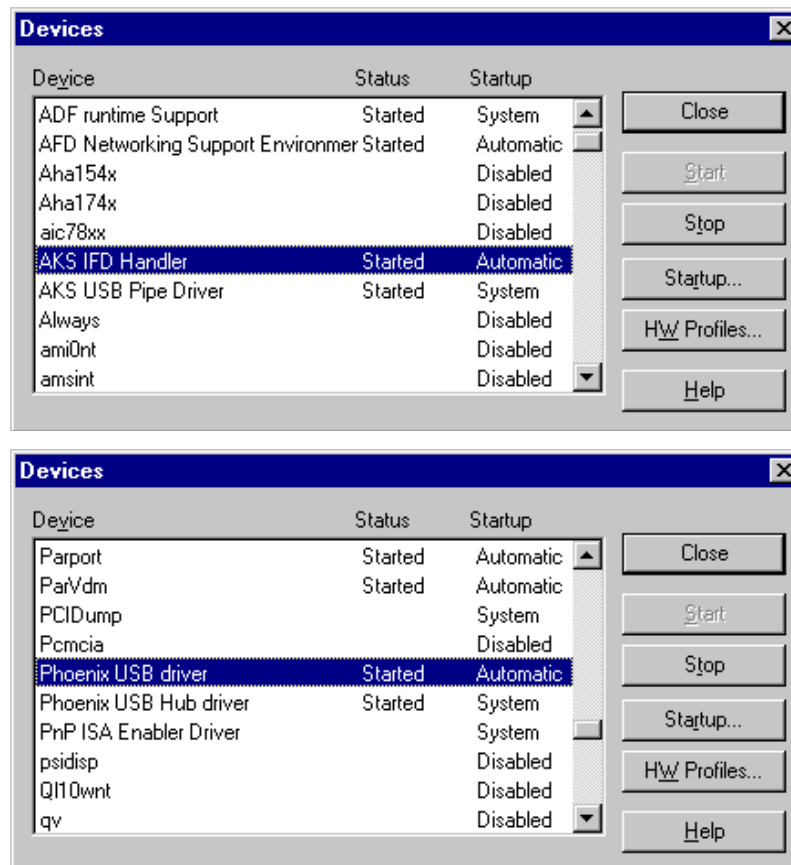
To check whether USB support is enabled for your Windows NT 4.0 system and whether the eToken RTE installation has been successful:

- 1 Open the Windows **Programs Menu**.
- 2 Select **Settings**.
- 3 Select Control Panel.
- 4 **Select Devices**. A list is displayed of the devices currently enabled in your system

USB support is enabled if the status of the following devices is shown as Started:

- **AKS IFD Handler**
- **AKS USB Pipe Driver**
- **Phoenix USB driver**
- **Phoenix USB Hub driver**

These are shown in the examples below:



If the Device list does not include these four entries, the installation failed. Reinstall or contact Technical Support.

Technical Support

If you are unable to solve the problems that you are experiencing and require technical support and assistance, please contact Aladdin by telephone, fax or email, as follows:

Tel: +972 3 636 2266 ext. 2

Fax: +972 3 537 5796

Email: etoken.techsup@Aladdin.com

Website: <http://aladdin.com/support>