

Entrust Authority™

# Security Manager Administration 8.1

## User Guide

Document issue: 1.0

Date of Issue: January 2012



Copyright © 2012 Entrust. All rights reserved.

Entrust is a trademark or a registered trademark of Entrust, Inc. in certain countries. All Entrust product names and logos are trademarks or registered trademarks of Entrust, Inc. in certain countries. All other company and product names and logos are trademarks or registered trademarks of their respective owners in certain countries.

This information is subject to change as Entrust reserves the right to, without notice, make changes to its products as progress in engineering or manufacturing methods or circumstances may warrant.

Export and/or import of cryptographic products may be restricted by various regulations in various countries. Export and/or import permits may be required.

## About this guide .....23

Revision information .....	24
Documentation conventions .....	25
Note and Attention text .....	25
Related documentation .....	26
Obtaining documentation .....	27
Documentation feedback .....	27
Obtaining technical assistance .....	28
Technical support .....	28
Email address.....	28
Professional Services .....	28

## Installing and uninstalling Security Manager Administration and the online help. 31

Installing Security Manager Administration .....	32
Upgrading Security Manager Administration .....	34
Uninstalling Security Manager Administration .....	35
Installing Security Manager Administration Online Help .....	36
Upgrading Security Manager Administration Online Help .....	37
Uninstalling Security Manager Administration Online Help .....	38

## Getting started in Security Manager Administration. ....39

Token support in Security Manager Administration .....	40
Using new and used tokens .....	40
Drivers for token readers .....	41
Creating your profile .....	42
Using Security Manager Administration with multiple Security Managers ..	44
Logging in to Security Manager Administration .....	46
Locking Security Manager Administration .....	49

Showing and hiding the Security Manager Administration toolbar	50
Changing your password	51
Authorizing sensitive operations	52
Configuring the Security Manager license information	55
Configuring Security Manager Administration preferences	57
Configuring general preferences	57
Configuring directory search preferences	59
Configuring search performance preferences	60
Configuring user list preferences	62
Configuring group information preferences	64
Recovering your profile	66

## **Using the Directory Browser . . . . .69**

Logging in to the Directory Browser	70
Locking the Directory Browser	71
Finding entries in the directory	72
Adding entries to the directory	74
Deleting directory entries	76
Adding attributes to directory entries	77
Adding directory attribute values	79
Replacing directory attribute values	80
Deleting directory attributes or attribute values	81
Changing the Security Manager Directory Administrator password	82

## **Setting up the Certification Authority. . . . .83**

Configuring the Security Policy	84
Configuring the Administration Policy	85
Interaction between CRL settings in the Security Policy	88
Configuring the encryption and verification OIDs	90
Adding OIDs	90
Deleting OIDs	92
Adding OIDs to the default policy list	93
Removing OIDs from the default policy list	94
Configuring queued requests	95
Configuring options for the various certificate categories	97

Exporting the current CA certificate .....	102
Viewing CA information .....	104
<b>Configuring users' key pairs .....</b>	<b>107</b>
Overview of key pairs .....	108
The dual-usage key pair .....	108
The encryption key pair .....	109
The signing key pair .....	110
The nonrepudiation key pair .....	110
The EFS encryption key pair .....	111
Supported key-pair models .....	113
1-key pair option .....	114
2-key pair option .....	115
3-key pair option .....	116
4-key pair option .....	117
Configuring key-pair users .....	118
Configuring V1 1-key-pair users .....	119
Creating 1-key-pair users .....	119
Changing 2-key-pair users into 1-key-pair users .....	120
Changing 1-key-pair users into 2-key-pair users .....	121
Configuring V2 key-pair users .....	122
Which path to choose? .....	122
Default configuration path .....	126
Customized configuration path .....	127
<b>Administering users .....</b>	<b>129</b>
User states .....	131
Email addresses in Security Manager .....	132
Including email addresses in distinguished names .....	132
Using special characters in user names .....	133
Finding users .....	134
Finding users by Entrust properties .....	134
Finding users by directory attributes .....	142
Creating new users .....	146
Adding existing users .....	152

Managing activation codes	153
Distributing activation codes	153
Configuring the lifetime of activation codes	154
Viewing activation codes and expiry dates	155
Reissuing activation codes	156
Creating user profiles	158
Recovering user key pairs	162
Setting users for key recovery	163
Canceling key recovery	164
Recovering user profiles	165
Restricting users	169
Deactivating users	169
Revoking user certificates	170
Suspending user certificates	171
Archiving and removing users	171
Deactivating and reactivating users	172
Deactivating users	172
Reactivating users	173
Revoking user certificates	174
Reasons for revoking certificates	174
Login messages received for revoked certificates	175
Revoking certificates	177
Issuing a new CRL after revoking a certificate	180
Suspending user certificates	181
Archiving and retrieving users	183
Archive files	183
Viewing archive files	185
Moving archive files	185
Deleting archive files	185
Archiving users	186
Retrieving archived users	186
Removing users from the database	191
Modifying distinguished names	192
Changing distinguished names	193

Troubleshooting DN changes. . . . .	197
Canceling DN changes . . . . .	198
Assigning new distinguished names . . . . .	199
Restoring user certificates to the directory . . . . .	201
Moving users to a new Certification Authority . . . . .	202
Moving users versus creating users . . . . .	203
Moving a user's decryption private keys . . . . .	203
Moving a user's certificates . . . . .	204
Summary of steps for moving users . . . . .	204
Establishing trust between the Certification Authorities . . . . .	206
Exchanging the CA public keys . . . . .	206
Exporting users . . . . .	210
Viewing import files . . . . .	212
Canceling user export operations . . . . .	213
Handling user information . . . . .	213
Importing users . . . . .	215
Logging in to the new CA . . . . .	216
Recovering the Entrust profile manually. . . . .	217
Completing user export operations . . . . .	217
Configuring user properties . . . . .	219
Configuring general user properties . . . . .	219
Configuring user certificate types . . . . .	221
Viewing and exporting user certificates . . . . .	223
Configuring user key update options . . . . .	230
Viewing DN change history . . . . .	237
Configuring user encryption and verification OIDs . . . . .	237
Updating key pairs . . . . .	240
Notifying client applications . . . . .	242
Changing user profiles . . . . .	243
Allowing profile export . . . . .	244
Overview of allowing profile export . . . . .	245
Roaming users and CAPI profile export. . . . .	245
Creating or modifying a user policy to allow profile export . . . . .	246
Assigning the user policy to a role . . . . .	246
Creating or modifying users to allow profile export . . . . .	246

Converting V2 users to V1 users .....	247
---------------------------------------	-----

## **Configuring subjectAltName values .....249**

Using the subjectAltName extension .....	250
SubjectAltName components .....	251
Configuring auto-population of the subjectAltName from the directory ..	257
Adding, modifying, or deleting subjectAltName component values .....	260
Updating subjectAltName component values from the directory .....	263
Viewing and exporting subjectAltName component values .....	267
Excluding the subjectAltName from certificate definitions .....	270
Setting the criticality of the subjectAltName extension .....	273

## **Performing bulk operations .....275**

Bulk command syntax .....	276
Commands and arguments .....	276
Backslash .....	276
Double quotation marks .....	277
Curly braces .....	277
Tcl output .....	278
Additional resources about Tcl .....	278
Creating bulk files .....	279
Processing bulk files .....	280
Viewing bulk output log files .....	283
Success messages .....	283
Failure messages .....	284
Advanced bulk processing .....	285
Adding users in bulk to Security Manager .....	288
Creating customized directory entries in bulk .....	296
Setting up users for key recovery in bulk .....	298
Creating the bulk file and performing key recovery .....	298
Canceling key recovery in bulk .....	299
Deactivating, reactivating, and deleting users in bulk .....	300
Deactivating users in bulk .....	300
Reactivating users in bulk .....	301
Deleting users from the directory in bulk .....	301



Revoking user certificates in bulk .....	303
Revoking user certificates in bulk .....	303
Putting user certificates on hold in bulk .....	305
Taking certificates off hold in bulk .....	306
Changing user information in bulk .....	308
Changing users' DN's in bulk .....	308
Canceling the change DN operation in bulk .....	313
Changing user properties in bulk .....	314
Adding and deleting directory attributes in bulk .....	323
Restoring information to the directory in bulk .....	325
Updating users' key pairs in bulk .....	326
Notifying client applications in bulk .....	327
Reissuing activation codes in bulk .....	328
<b>Administering groups .....</b>	<b>329</b>
Viewing groups .....	330
Creating groups .....	332
Adding members to groups .....	334
Removing members from groups .....	336
Renaming groups .....	338
Deleting groups .....	339
<b>Administering searchbases .....</b>	<b>341</b>
Viewing searchbases .....	342
Adding searchbases .....	344
Adding searchbases to the directory .....	345
Adding searchbases to Security Manager .....	347
Modifying searchbases .....	349
Deleting searchbases .....	351
<b>Administering roles .....</b>	<b>353</b>
Predefined Security Manager user roles .....	354
Viewing roles .....	358
Creating roles .....	361
Modifying roles .....	363

Checking permission dependencies .....	368
Deleting roles .....	370
Permissions reference .....	371
Audit log permissions .....	372
Bulk operations and reports permissions .....	373
Certificate permissions .....	373
Certificate categories permissions. ....	373
Certificate types permissions .....	374
Certification Authority (CA) permissions .....	374
CA permissions .....	375
Cross-certified CA permissions .....	375
Subordinate CA permissions. ....	376
Directory permissions .....	376
Extended Access Control (CVCA) permissions .....	376
Anchor CVCA permissions .....	377
DV permissions .....	377
Foreign CVCA permissions. ....	378
Extended Access Control (DV) permissions .....	379
Anchor DV permissions .....	379
CVCA permissions .....	380
Inspection System permissions .....	381
Group permissions .....	382
License information permissions .....	382
Policy OIDs permissions .....	382
Queued requests permissions .....	383
Role permissions .....	384
Searchbase permissions .....	384
Security policy permissions .....	385
User template permissions .....	386
User permissions .....	386
General user permissions .....	387
Advanced user permissions .....	387
Other user permissions .....	388

## **Administering user policies. ....391**

User policy overview .....	392
----------------------------	-----

Predefined user policies . . . . .	393
Viewing user policies . . . . .	395
Viewing policy certificates . . . . .	397
Creating user policies . . . . .	398
Modifying user policies . . . . .	402
Mapping policy certificates to certificate definitions . . . . .	404
Deleting user policies . . . . .	405
Importing and exporting user policies . . . . .	406
Client policy attributes reference . . . . .	407
Policy Certificate expires in (days) . . . . .	407
Password expires in (weeks) . . . . .	407
Password history . . . . .	407
Password length (characters) . . . . .	408
Password needs non-alpha char. . . . .	408
Password needs uppercase letter . . . . .	408
Password needs lowercase letter . . . . .	408
Password needs number . . . . .	409
Disable single login . . . . .	409
Login timeout (minutes) . . . . .	409
Symmetric encryption algorithms . . . . .	410
Key type for signatures . . . . .	411
Key type for encryption . . . . .	412
Message in Entrust-Ready clients . . . . .	412
Permit roaming . . . . .	413
Permit desktop . . . . .	413
Enforce token usage . . . . .	413
Allow personal addr. book use. . . . .	414
Allow CA personal addr. book use. . . . .	414
Permit Server Login usage . . . . .	414
Enforce identity usage . . . . .	414
DN encoding formats . . . . .	414
Perform dir. consistency check. . . . .	415
Management Client . . . . .	416
Force Original CD Compliance . . . . .	416
Enforce S/MIME . . . . .	416
Allow S/MIME Secure Receipts . . . . .	416

Acceptable policy OIDs . . . . .	417
Do not process policy mappings. . . . .	418
Require policy . . . . .	418
PKIX compliance . . . . .	418
FPKI compliance . . . . .	419
Do not process anyPolicy. . . . .	420
HTTP Proxy for CRL Requests . . . . .	420
Allow PKCS#12 Export. . . . .	421
All Exportable . . . . .	421
Minimum PKCS#12 Hash Count . . . . .	421
Enable CAPI Synchronization . . . . .	422
Unprotected CAPI key storage? . . . . .	422
Private key export from CAPI? . . . . .	423
Number of key pairs . . . . .	423
Prevent single login register . . . . .	424
Delay single login register . . . . .	424
Maximum bad login attempts . . . . .	424
Login attempt window. . . . .	425
ICE settings signed. . . . .	425
ICE settings ignored. . . . .	425
Enable the use of an ARL cache . . . . .	425
Enable the use of a CRL cache . . . . .	425
Enable the use of a XCert cache. . . . .	425
Enable the use of a Cert cache . . . . .	425
Secure Delivery SMTP . . . . .	426
Content Scanner SMTP . . . . .	426
Express Search Source Order . . . . .	426
Check e-mail on verification. . . . .	427
Check e-mail on encryption . . . . .	427
Cross Cert DNs . . . . .	427
Auto-Associate MS Outlook . . . . .	427
Searchbase Search Order . . . . .	427
CRL grace period . . . . .	428
CRL grace percentage . . . . .	428
Reg. Pwd Max Fail. . . . .	428
Reg. Client type . . . . .	428

Self-admin policy . . . . .	429
Public Token Certs . . . . .	429
Enforce protected key transfer . . . . .	429
Allow Spillover File for Tokens . . . . .	429
Messaging Server SMTP . . . . .	430
Symmetric Key Access . . . . .	430
Algorithm for profile protection . . . . .	430
Allow Self Revocation . . . . .	431
Certificate definition policy attributes reference . . . . .	432
Policy Certificate expires in (days) . . . . .	432
Certificate lifetime . . . . .	432
Certificate lifetime (Days) . . . . .	433
Private key usage period . . . . .	433
Ignore per user lifetime . . . . .	434
Publishing policy . . . . .	434
Publish revoked certs. . . . .	434
Publish expired certs. . . . .	435
Publishing DN . . . . .	435
Publish at key update . . . . .	435
Exclude privateKeyUsagePeriod. . . . .	436
Exclude basicConstraints . . . . .	436
Exclude CDP . . . . .	436
Exclude entrustVersInfo. . . . .	436
Exclude subjectKeyId. . . . .	437
Exclude subjectAltName . . . . .	437
Exclude certificatePolicy . . . . .	437
Exclude subjectAltName parts . . . . .	437
SubjectAltName criticality . . . . .	438
Enable CMP override. . . . .	438
Allow unknown extensions . . . . .	439
Enforce client policy. . . . .	439
Only latest key can sign CMP . . . . .	439
Key can sign CMP . . . . .	439
Use CMP publish flag . . . . .	439
Algorithm for key pair . . . . .	440
Back up private key . . . . .	441

Generate key at client . . . . .	441
Key usage policy . . . . .	441
CSP to manage keys . . . . .	442
Protect key storage for CSP . . . . .	442
Private key export from CSP . . . . .	442
Symmetric Key Access . . . . .	442
Force client key generation in CSP . . . . .	443
Update cert. at % of lifetime . . . . .	443
Enable cert. update date . . . . .	443
Cert. update date. . . . .	443
Certificate Signing Alg . . . . .	444
Revoke superseded certs.. . . . .	444

## **Modifying the user template file and user types. . . . .447**

Overview of user template file and user types . . . . .	448
Purpose of user types . . . . .	448
Available user types . . . . .	449
Summary of steps . . . . .	450
Exporting the template definition file . . . . .	451
Customizing the New User dialog box . . . . .	452
Adding a new user type . . . . .	452
Adding attributes to user types . . . . .	455
Determining the structure of the common name for the Person user type	
459	
Activating Hardware and Person 4.0 PKI user types . . . . .	459
Importing the template definition file . . . . .	461
Testing the template definition file . . . . .	462

## **Cross-certifying with other CAs. . . . .465**

About cross-certification . . . . .	467
Establishing trust using cross-certification . . . . .	468
One-way trust versus two-way trust . . . . .	469
How client applications use cross-certificates . . . . .	469
How Bob encrypts a file for Alice . . . . .	470
Key identifiers in cross-certification . . . . .	470

Changing the format of key identifiers .....	471
Methods for establishing trust .....	473
Cross-certifying online .....	474
Online cross-certification requirements .....	474
Communication between CAs .....	475
Performing online cross-certification .....	475
Cross-certifying offline .....	484
Offline cross-certification requirements .....	484
Performing offline cross-certification .....	484
Viewing cross-certificates .....	495
Removing cross-certificates from the directory .....	497
Publishing cross-certificates to the directory .....	499
Revoking cross-certificates .....	501
Taking cross-certificates off hold .....	504
Updating cross-certificates .....	505
Setting policy constraints requirements in protocol certificates .....	506
<b>Administering subordinate CAs .....</b>	<b>509</b>
Creating subordinate CA certificates .....	510
Creating subordinate CA certificates online .....	510
Creating subordinate CA certificates offline .....	513
Revoking subordinate CA certificates .....	518
Taking subordinate CA certificates off hold .....	520
Exporting subordinate CA certificates .....	521
Preparing the directory .....	523
Chaining and referring directories .....	523
<b>Customizing certificates. ....</b>	<b>525</b>
About certificates .....	527
Security Manager certificate categories .....	527
About the X.509 recommendation .....	528
Dividing certificate categories into certificate types .....	529
Why customize certificates? .....	530
Customizing content for certificates .....	531
Working with Security Manager certificate specifications .....	531

Contents of the master.certspec file . . . . .	532
Predefined certificate types . . . . .	535
Predefined Enterprise certificate types . . . . .	535
Predefined Web certificate types . . . . .	538
Working with the master.certspec file . . . . .	539
Creating the master.certspec file . . . . .	541
Opening the master.certspec file . . . . .	541
Navigating the master.certspec file . . . . .	542
Processing changes to the master.certspec file . . . . .	543
Customizing Enterprise and Web certificates . . . . .	544
Creating and modifying user certificate information . . . . .	545
What are V1 and V2 certificates? . . . . .	546
What is a certificate type description? . . . . .	546
What is a certificate definition? . . . . .	547
What is a certificate extension? . . . . .	547
Using V1 certificate type descriptions and certificate extensions	548
Using V2 certificate definitions and certificate extensions. . . .	548
Editing a certificate type description . . . . .	549
Editing certificate definitions . . . . .	551
Editing certificate extensions . . . . .	553
Extensions for V1 certificate types . . . . .	553
Associating a certificate extension with a V1 certificate type . .	554
Extensions for V2 certificate types . . . . .	555
Associating a certificate extension with a V2 certificate type . .	556
Defining a certificate extension . . . . .	557
Extn Name. . . . .	558
Extn OID . . . . .	558
Extn Crit . . . . .	559
Extn Opt . . . . .	560
Extn Type. . . . .	561
Extn Value . . . . .	563
Editing a variable . . . . .	565
Setting up default values . . . . .	567
Extended Key Usage (EKU) certificate extensions . . . . .	568
Certificate specification examples . . . . .	569



Customizing policy certificates .....	571
Creating and modifying a certificate type .....	572
What is a certificate type description? .....	573
What is a certificate attribute? .....	573
Editing a certificate type description .....	573
Editing certificate attributes .....	575
Defining a certificate attribute in a subsection .....	576
Attr Name .....	577
Attr OID .....	577
Attr Type .....	577
Attr Value .....	578
Editing a variable for a certificate attribute .....	578
Enforcing the use of custom policy certificates .....	578
Customizing cross-certificates .....	580
Creating and modifying a cross-certificate type .....	580
What is a certificate type description? .....	581
What is a certificate extension? .....	581
Using certificate type descriptions and certificate extensions ..	581
Adding a certificate type description .....	582
Editing certificate extensions .....	584
Defining a certificate extension in a subsection .....	584
Applications of certificate extensions in cross-certificates. ....	585
Limiting trust using distinguished names .....	586
Limiting trust using policy settings .....	587
Limiting trust using CA domains .....	588
Examples of cross-certificate types .....	589
Customizing CA certificates .....	591
Excluding certificate extensions by certificate type .....	593
Extensions you can exclude from user certificates .....	594
Extensions you can exclude from CA certificates .....	594
Extensions you can exclude from cross-certificates .....	595
Excluding default extensions from an X.509 certificate .....	596
Making a certificate type obsolete .....	599
Defining a replacement certificate type .....	599

Customizing database fields .....	601
Defining a database field .....	602
Example of database fields .....	603
Turning off revocation checking for foreign certificates .....	605

## **Working with audit logs and creating reports .....607**

Viewing Security Manager Administration log files .....	608
Working with audit logs .....	609
Viewing audit logs .....	609
Viewing audit log details .....	611
Printing audit logs .....	612
Description of audit log fields .....	612
Sorting audit logs .....	613
Clearing audit logs .....	614
Saving audit logs .....	614
Creating a debug log .....	615
Producing reports in Security Manager Administration .....	616
Report contents .....	616
Report formats .....	617
XML-format reports.....	617
Tab-delimited reports.....	618
Creating reports .....	619
Report fields .....	629

## **Administering attribute certificates .....635**

About attribute certificates .....	636
How are attribute certificates used? .....	636
Examples of using attribute certificates .....	637
Attribute certificates and Security Toolkit for the Java Platform ...	637
About Security Toolkit for Java Platform .....	637
Steps for working with attribute certificates .....	638
Creating custom attribute certificate types .....	639
Issuing attribute certificates .....	640
Managing attribute certificates .....	643
Reissuing attribute certificates .....	643

Automatic reissue of attribute certificates .....	643
Deleting attribute certificates .....	645
Replacing attribute certificates .....	647
Viewing the contents of attribute certificates .....	648
Saving attribute certificates to file .....	648
Administering users with attribute certificates .....	650

## **Bulk commands reference ..... 651**

Authorizing operations in bulk files .....	652
Group bulk commands .....	653
group .....	653
group_create .....	653
group_delete .....	654
group_removeallusers .....	654
Searchbase bulk commands .....	655
searchbase .....	655
searchbase_add .....	656
searchbase_delete .....	656
searchbase_setdn .....	656
searchbase_setlabel .....	657
searchbase_userscansearch .....	657
User bulk commands .....	658
user .....	659
user_add .....	659
user_addtodn .....	660
user_applyproperties .....	660
user_archive .....	660
user_assigndn .....	661
user_cancelchangedn .....	661
user_cancelhold .....	661
user_cancelrecover .....	662
user_convertv1 .....	662
user_createdirentry .....	662
user_deactivate .....	663
user_deletedirentry .....	663
user_export .....	664

Starting a user export operation . . . . .	664
Finishing the user export operation . . . . .	665
Canceling a user export operation . . . . .	665
user_free . . . . .	666
user_import . . . . .	666
user_notifyclient . . . . .	666
user_reactivate . . . . .	667
user_recover . . . . .	667
user_reissueactivationcodes . . . . .	667
user_renamedirentry . . . . .	668
user_restoretodir . . . . .	668
user_retrieve . . . . .	668
user_revoke . . . . .	669
user_setattribute . . . . .	670
user_setparentdn . . . . .	670
user_setproperty . . . . .	671
Setting certificate category and type . . . . .	671
Setting group membership . . . . .	672
Setting the subjectAltName (alternate identity) . . . . .	672
Setting key update options—key lifetimes . . . . .	673
Setting key update options—key expiry . . . . .	673
Setting certificate extension variable values . . . . .	674
Setting user role . . . . .	675
Setting encryption OIDs . . . . .	675
Setting verification OIDs . . . . .	676
user_settemplate . . . . .	677
user_updatekeypairs . . . . .	677
user_refresh_subaltname_from_dir . . . . .	678

## **Entering international characters in distinguished names . . . . .679**

What is internationalization support? . . . . .	680
When to use international characters . . . . .	681
Attributes that support international characters . . . . .	681
Using international characters during installation . . . . .	681
Directory issues with international characters . . . . .	682

How to enter international characters in Security Manager Administration	683
Entering international characters from an ASCII keyboard	683
Entering international characters using the IME	684
How Security Manager Administration displays international characters	685
<b>Glossary of terms</b>	<b>687</b>
<b>Index</b>	<b>703</b>



## About this guide

This book is part of the documentation suite included with Entrust Authority Security Manager Administration. It includes information about using Entrust Authority Security Manager Administration, the graphical interface for administering Security Manager.

---

**Note:** This guide does not contain information for users of Entrust desktop applications. Consult the documentation that accompanies each application.

---

Topics in this section:

- [“Revision information” on page 24](#)
- [“Documentation conventions” on page 25](#)
- [“Related documentation” on page 26](#)
- [“Obtaining documentation” on page 27](#)
- [“Obtaining technical assistance” on page 28](#)

# Revision information

**Table 1:** Revisions in this document

Document issue and date	Section	Description
1.0 January 2012	All sections	First release of this guide.



# Documentation conventions

Following are documentation conventions which appear in this guide:

**Table 2:** Typographic conventions

Convention	Purpose	Example
<b>Bold</b> text (other than headings)	Indicates graphical user interface elements and wizards.	Click <b>Next</b> .
<i>Italicized</i> text	Used for book or document titles.	<i>Entrust IdentityGuard Administration Guide</i>
<a href="#">Blue</a> text	Used for hyperlinks to other sections in the document.	Entrust TruePass supports the use of many types of <a href="#">digital IDs</a> .
<a href="#">Underlined blue</a> text	Used for Web links.	For more information, visit our Web site at <a href="http://www.entrust.com">www.entrust.com</a> .
Courier type	Indicates installation paths, file names, Windows registry keys, commands, and text you must enter.	Use the <code>entrust-configuration.xml</code> file to change certain options for Verification Server.
Angle brackets < >	Indicates variables (text you must replace with your organization's correct values).	By default, the <code>entrust.ini</code> file is located in <code>&lt;install_path&gt;/conf/security/entrust.ini</code> .
Square brackets <code>[courier type]</code>	Indicates optional parameters.	<code>dsa passwd [-ldap]</code>

## Note and Attention text

Throughout this guide, there are paragraphs set off by ruled lines above and below the text. These paragraphs provide key information with two levels of importance, as shown below.

---

**Note:** Information to help you maximize the benefits of your Entrust product.

---

---

**Attention:** Issues that, if ignored, may seriously affect performance, security, or the operation of your Entrust product.

---

# Related documentation

This section provides a list of useful reference material. Some of these documents are also mentioned throughout this guide in relevant places as related reading material.

- *Entrust Authority Security Manager Release Notes*
- *Entrust Authority Security Manager Deployment Guide*
- *Entrust Authority Security Manager Directory Configuration Guide*
- *Entrust Authority Security Manager Installation Guide*
- *Entrust Authority Security Manager Operations Guide*
- *Entrust Authority Security Manager Administration Release notes*
- *Entrust Authority Security Manager Administration User Guide*
- *Entrust Authority Security Manager Administration Online Help*
- *Entrust ePassport Solutions Guide*

# Obtaining documentation

Entrust product documentation, white papers, technical notes, and a comprehensive Knowledge Base are available through Entrust TrustedCare Online. If you are registered for our support programs, you can use our Web-based Entrust TrustedCare Online support services at:

<https://secure.entrust.com/trustedcare>

## Documentation feedback

You can rate and provide feedback about Entrust product documentation by completing the online feedback form. You can access this form by

- clicking the *Report any errors or omissions* link located in the footer of Entrust's PDF documents (see bottom of this page).
- following this link: <http://www.entrust.com/products/feedback/index.cfm>

Feedback concerning documentation can also be directed to the Customer Support email address.

[support@entrust.com](mailto:support@entrust.com)

# Obtaining technical assistance

Entrust recognizes the importance of providing quick and easy access to our support resources. The following sections provide details about the technical support and professional services available to you.

## Technical support

Entrust offers a variety of technical support programs to help you keep Entrust products up and running. To learn more about the full range of Entrust technical support services, visit our Web site at:

<http://www.entrust.com/>

If you are registered for our support programs, you can use our Web-based support services.

Entrust TrustedCare Online offers technical resources including Entrust product documentation, white papers and technical notes, and a comprehensive Knowledge Base at:

<https://secure.entrust.com/trustedcare>

If you contact Entrust Customer Support, please provide as much of the following information as possible:

- your contact information
- product name, version, and operating system information
- your deployment scenario
- description of the problem
- copy of log files containing error messages
- description of conditions under which the error occurred
- description of troubleshooting activities you have already performed

## Email address

The email address for Customer Support is:

[support@entrust.com](mailto:support@entrust.com)

## Professional Services

The Entrust team assists e-businesses around the world to deploy and maintain secure transactions and communications with their partners, customers, suppliers and employees. We offer a full range of professional services to deploy our e-business solutions successfully for wired and wireless networks, including planning and design,

installation, system integration, deployment support, and custom software development.

Whether you choose to operate your Entrust solution in-house or subscribe to hosted services, Entrust Professional Services will design and implement the right solution for your e-business needs. For more information about Entrust Professional Services please visit our Web site at:

<http://www.entrust.com>



# Installing and uninstalling Security Manager Administration and the online help

This chapter describes how to install, upgrade, and uninstall Security Manager Administration and the Security Manager Administration online help.

---

**Note:** You can only install Security Manager Administration and the Security Manager online help on a Microsoft Windows operating system.

---

This chapter includes the following sections:

- [“Installing Security Manager Administration” on page 32](#)
- [“Upgrading Security Manager Administration” on page 34](#)
- [“Uninstalling Security Manager Administration” on page 35](#)
- [“Installing Security Manager Administration Online Help” on page 36](#)
- [“Upgrading Security Manager Administration Online Help” on page 37](#)
- [“Uninstalling Security Manager Administration Online Help” on page 38](#)

# Installing Security Manager Administration

This section describes how to install Security Manager Administration. You can install Security Manager Administration on the same server where you installed Security Manager, or locally on a separate computer used for administrative purposes.

If you are not installing Security Manager Administration on the Security Manager server, save a copy of the Security Manager `entrust.ini` and `entrustra.ini` files to any folder on the Security Manager Administration computer.

## To install Security Manager Administration

- 1** Log in to the computer where you will install Security Manager Administration.  
Any user with administrative permissions for the computer can install Security Manager Administration.
- 2** Go to the Entrust TrustedCare Online Web site and download the Security Manager Administration installer (`SMA_81SP1_setup.exe`).
- 3** If you are not installing Security Manager Administration on the Security Manager server, save a copy of the Security Manager `entrust.ini` and `entrustra.ini` files to any folder.
- 4** Double-click the installer file to run the Security Manager Administration installer.
- 5** If you are installing Security Manager Administration on a different server than Security Manager:
  - The **Location of `entrust.ini` file** dialog box appears.
  - a** Select the folder where you saved the `entrust.ini` file and then click **OK**.  
The **Location of `entrustra.ini` file** dialog box appears.
  - b** Select the folder where you saved the `entrustra.ini` file and then click **OK**.  
The **InstallShield Wizard** appears.
- 6** If you are installing Security Manager Administration on the same server as Security Manager:
  - The **Entrust.ini File Found** dialog box appears.
  - a** To use the existing `entrust.ini` file, click **Yes**. To use a different `entrust.ini` file, click **No**.
  - b** If you chose to use a different `entrust.ini` file, the **Location of `entrust.ini` file** dialog box appears.  
Select the folder where you saved the `entrust.ini` file and then click **OK**.
  - c** The **Entrustra.ini File Found** dialog box appears.  
To use the existing `entrustra.ini` file, click **Yes**. To use a different `entrustra.ini` file, click **No**.



- d** If you chose to use a different `entruststra.ini` file, the **Location of entruststra.ini file** dialog box appears.

Select the folder where you saved the `entruststra.ini` file and then click **OK**.

The **InstallShield Wizard** appears.

- 7** Click **Next** to continue.

The **License Agreement** page appears.

- 8** Click **Yes** to accept the license agreement.

The **Choose Destination Location** page appears.

- 9** Click **Browse** to choose a different install folder. Click **Next** to install Security Manager Administration.

After Security Manager Administration installs, the **InstallShield Wizard Complete** page appears.

- 10** Click **Finish**.

The installer saves a copy of the `entrust.ini` file into the Security Manager Administration installation directory, typically:

`C:\Program Files (x86)\Entrust\Security Manager Administration`

The installer also saves a copy of the `entruststra.ini` file into the following folder:

`C:\ProgramData\Entrust\Security Manager Administration`

# Upgrading Security Manager Administration

This section describes how to upgrade your existing Security Manager Administration installation to Security Manager Administration 8.1.

## To upgrade to Security Manager Administration 8.1

- 1** Download the Security Manager Administration installer (SMA\_81SP1\_setup.exe) from Entrust TrustedCare.
- 2** If Security Manager Administration is open, close it to prevent conflicts during the upgrade.
- 3** Run the installer to start the upgrade.  
The **Upgrade?** dialog box appears, asking you if you want to upgrade Security Manager Administration.
- 4** Click **Yes** to upgrade Security Manager Administration.  
The InstallShield Wizard appears.
- 5** Click **Next**.  
The **License Agreement** page appears.
- 6** Read the license agreement and then click **Yes** to accept the license agreement.  
The **Setup Status** page appears, showing the progress of the upgrade. When the upgrade completes, the **InstallShield Wizard Complete** page appears.
- 7** Click **Finish** to exit the wizard.

# Uninstalling Security Manager Administration

This section describes how to uninstall Security Manager Administration. Use this procedure to uninstall Security Manager Administration from a computer where you no longer use Security Manager Administration.

## To uninstall Security Manager Administration

- 1** Log in to the computer hosting Security Manager Administration as a user with administrative permissions.  
Any user with administrative permissions for the computer can remove Security Manager Administration.
- 2** Open the Windows Control Panel and then open **Add or Remove Programs**.
- 3** Select **Entrust Authority(TM) Security Manager Administration 8.1 SP1** and then click **Change/Remove**.  
The InstallShield Wizard Welcome window appears displaying options to modify, repair, or remove the program.
- 4** Click **Remove**, then click **Next**.  
The **Confirm Uninstall** dialog box appears.
- 5** Click **OK** to remove the program.  
The InstallShield window displays the progress as the program is removed. When removal is complete, the **Maintenance Complete** dialog box appears.
- 6** Click **Finish** to complete the removal and close the InstallShield window.
- 7** To completely remove Security Manager Administration from an administrative computer separate from the server hosting Security Manager, you must also delete the `entrust.ini` and `entrust.ra.ini` files, and all administrator profiles (EPF files).

# Installing Security Manager Administration Online Help

This section describes how to install the Security Manager Administration online help to support Security Manager Administration. You should install Security Manager Administration online help on each computer where you have installed Security Manager Administration.

## To install Security Manager Administration online help

- 1** Log in to the computer where you installed Security Manager Administration.  
Any user with administrative permissions for the computer can install the Security Manager Administration online help.
- 2** Go to the Entrust TrustedCare Online Web site and download the Security Manager Administration online help installer (*SMA\_81SP1\_Help.exe*).
- 3** Double-click the installer file to run the Security Manager Administration online help installer.  
The **InstallShield Wizard** appears.
- 4** Click **Next** to continue.  
The **Choose Destination Location** page appears.
- 5** Click **Browse** to choose a different install folder. Click **Next** to install the Security Manager Administration online help.  
After Security Manager Administration online help installs, the **InstallShield Wizard Complete** page appears.
- 6** Click **Finish**.

# Upgrading Security Manager Administration Online Help

Security Manager Administration Online Help is a unique component of Security Manager Administration. You cannot upgrade the online help. To upgrade the online help, you must uninstall the existing online help and then install the new version of the online help.

## To upgrade Security Manager Administration Online Help

- 1** Uninstall the existing Security Manager Administration Online Help (see [“Uninstalling Security Manager Administration Online Help” on page 38](#)).
- 2** Install the new version of Security Manager Administration Online Help (see [“Uninstalling Security Manager Administration Online Help” on page 38](#)).

# Uninstalling Security Manager Administration Online Help

This section describes how to uninstall Security Manager Administration Online Help. Use this procedure to uninstall Security Manager Administration Online help from a computer where:

- you must remove an existing Security Manager Administration online help in order to upgrade Security Manager Administration and the online help to a newer version
- you no longer use Security Manager Administration and Security Manager Administration online help

## To uninstall Security Manager Administration online help

- 1** Log in to the Windows server hosting Security Manager Administration online help as a user with administrative permissions.  
Any user with administrative permissions for the computer can remove Security Manager Administration online help.
- 2** Open the Windows Control Panel and then open **Add or Remove Programs**.
- 3** Select **Entrust Authority Security Manager Administration Online Help** and click **Change/Remove**.  
The InstallShield Wizard Welcome window appears displaying options to modify, repair, or remove the program.
- 4** Select **Remove**, then click **Next**.  
The **Confirm Uninstall** dialog box appears.
- 5** Click **OK** to remove the program.  
The InstallShield window displays the progress as the program is removed. When removal is complete, the **Maintenance Complete** dialog box appears.
- 6** Click **Finish** to complete the removal and close the InstallShield window.

# Getting started in Security Manager Administration

Security Manager Administration is the graphical interface that Entrust PKI administrators use to access and administer Entrust PKI data. Entrust PKI administrators are Entrust users with an administrative role. The data that Entrust PKI administrators can access and manage depends on their role. For more information about roles, see [“Administering roles” on page 353](#).

This chapter provides information about how to get started in Security Manager Administration.

This chapter includes the following sections:

- [“Token support in Security Manager Administration” on page 40](#)
- [“Creating your profile” on page 42](#)
- [“Using Security Manager Administration with multiple Security Managers” on page 44](#)
- [“Logging in to Security Manager Administration” on page 46](#)
- [“Locking Security Manager Administration” on page 49](#)
- [“Showing and hiding the Security Manager Administration toolbar” on page 50](#)
- [“Changing your password” on page 51](#)
- [“Authorizing sensitive operations” on page 52](#)
- [“Configuring the Security Manager license information” on page 55](#)
- [“Configuring Security Manager Administration preferences” on page 57](#)
- [“Recovering your profile” on page 66](#)

# Token support in Security Manager Administration

Security Manager Administration supports only PKCS#11 v2.01 (called v2 throughout this document) tokens. When using tokens, keep the following information in mind:

- When you use a token to log in to Security Manager Administration, that card must remain in the reader for Security Manager Administration to operate. You can remove other tokens used for authorization from the reader after the completion of the operation you are authorizing.
- You can connect only two token readers to the Security Manager Administration machine.
- Security Manager Administration supports only one vendor library at a time. For example, this means that you can use either Schlumberger or DataKey tokens, but not both.
- Security Manager Administration supports only one Entrust digital ID on a token at a given time.

For the most recent information about supported tokens and middleware, see the [Entrust Platform Support and Integration Center](#).

## Using new and used tokens

You can save a digital ID (also called a profile) to a new or used PKCS#11 v2 token. If you are using a token that already holds a profile, Security Manager Administration asks for confirmation before overwriting the existing profile.

Depending on the token vendor, you may need to initialize a new token before you can create a profile on it. Consult the documentation for specific token for more information.

In Security Manager Administration, a profile on a token is identified by the slot number of the token reader and the name assigned to the profile. Token readers (according to the policies of their vendors) determine the slot numbers.

---

**Note:** Security Manager Administration may not be able to distinguish between token profiles that have the same name and the same Entrust Support Files directory. Do not create token profiles for which these values are identical.

---



## Drivers for token readers

Pointers to the drivers for the token readers are kept in libraries identified by the Security Manager INI files. In most cases, token drivers and their library pointers are loaded automatically and do not need any administration.

However, you may need to add settings to the `entrust.ini` file under either of these circumstances:

- if you are using token readers that do not update the file
- if you are using hardware devices for Security Manager on the same machine as Security Manager Administration (see the *Security Manager Operations Guide*)

# Creating your profile

Before you can use Security Manager Administration, you must have a valid profile (EPF file). If you are the First Officer, your profile (typically `First Officer.epf`) was created when Security Manager was initialized.

If you do not have a profile, you must create a profile before you can log in to Security Manager Administration. To create a profile, another administrator must add you to Security Manager, generating a reference number and authorization code (see [“Creating new users” on page 146](#)). You use the reference number and authorization code to create your profile.

---

**Note:** If you plan to use hardware tokens with Security Manager, see [“Token support in Security Manager Administration” on page 40](#).

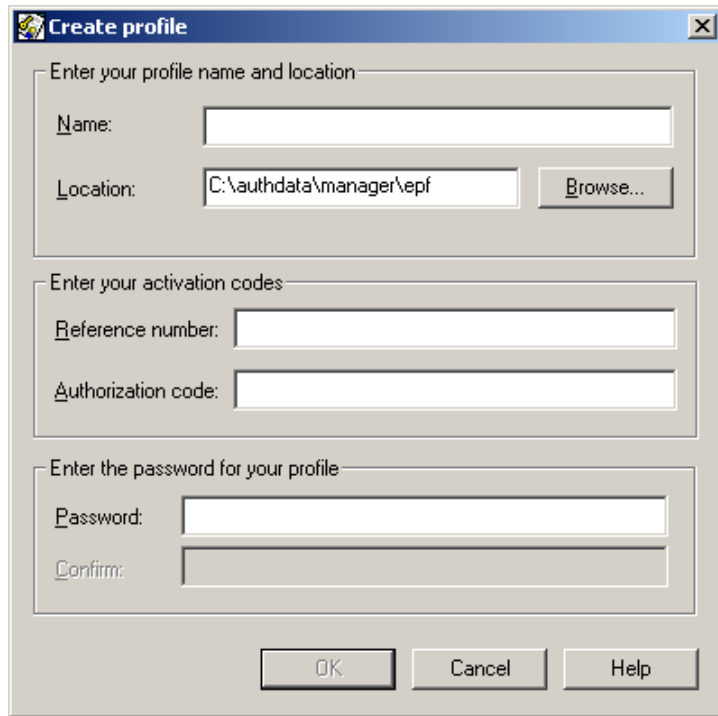
---

## To create an Entrust profile using Security Manager Administration

- 1 Obtain the reference number and authorization code from another Entrust PKI administrator. You need these codes to create the profile.
- 2 Install Security Manager Administration (see [“Installing Security Manager Administration” on page 32](#)).
- 3 Select **Start > All Programs > Entrust > Security Manager Administration**.  
The **Entrust Authority (TM) Security Manager Administration Log in** dialog box appears.



- 4 Click **Create Profile**.  
The **Create profile** dialog box appears.

A screenshot of a Windows-style dialog box titled "Create profile". The dialog box has a blue title bar with a close button (X) in the top right corner. It contains three main sections, each with a label and input fields. The first section is labeled "Enter your profile name and location" and contains a "Name:" label with an empty text box, and a "Location:" label with a text box containing "C:\authdata\manager\epf" and a "Browse..." button. The second section is labeled "Enter your activation codes" and contains a "Reference number:" label with an empty text box, and an "Authorization code:" label with an empty text box. The third section is labeled "Enter the password for your profile" and contains a "Password:" label with an empty text box, and a "Confirm:" label with an empty text box. At the bottom of the dialog box are three buttons: "OK", "Cancel", and "Help".

- 5 In the **Name** field, enter your name.  
Your profile will use the name you enter as its file name. For example, if you enter New Admin, your profile file name is New Admin.epf.
- 6 In the **Location** field, enter the location where you want to store your Entrust profile, or click **Browse** to select a location.
- 7 In the **Reference number** field, enter the reference number that you obtained earlier.
- 8 In the **Authorization code** field, enter the authorization code that you obtained earlier.
- 9 Enter a password for your profile in the **Password** and **Confirm** fields.
- 10 Click **OK**.

A dialog box appears stating that your profile password has been created successfully or that your password violates the applicable password rules.

Once your password is created successfully, the main Security Manager Administration window appears. You have created your profile and are now logged in to Security Manager Administration.

# Using Security Manager Administration with multiple Security Managers

You can set up Security Manager Administration so that it accesses different Security Managers, depending on the Entrust PKI administrator profile used to log in.

## To use Security Manager Administration with multiple Security Managers

- 1 Install Security Manager Administration using the `entrust.ini` and `entruststra.ini` files from one Security Manager (see [“Installing Security Manager Administration” on page 32](#)).
- 2 Copy the `entrust.ini` files from all Security Managers to the computer hosting Security Manager Administration.
- 3 In each Security Manager, create an Entrust PKI administrator, and copy the administrator’s profiles to the computer running Security Manager Administration. For instructions about creating users, see [“Creating new users” on page 146](#).
- 4 Open the `entruststra.ini` file that Security Manager Administration uses with a text editor.
- 5 Add a section called `[entrust ini override]`.
- 6 Add the following information to this section:

```
<profile_name>=<entrust.ini_name>
```

Where:

- `<profile_name>` is the full path to the Entrust PKI administrator profile.
- `<entrust.ini_name>` is the full path to the `entrust.ini` file for that particular Security Manager instance.

For example, if you are adding an administrator with a desktop profile:

```
C:\profile1\Officer1.epf=C:\profile1\entrust.ini
```

If you are adding an administrator with a profile stored on a token, point to the location where the auxiliary files (Entrust Support files) are stored on the computer (this is also the token profile path). For example:

```
C:\profile2\Officer2.tkn=C:\profile2\entrust.ini
```

---

**Note:** In this example, the file `Officer2.tkn` is a pointer that tells Security Manager Administration in what directory it can find the auxiliary files and `entrust.ini` file. There is not actually a file called `Officer2.tkn` present anywhere on the computer. For more information, see [“Token support in Security Manager Administration” on page 40](#).

---

- 7** Add as many paths as required.
- 8** Save your changes.

# Logging in to Security Manager Administration

To use Security Manager Administration, you must log in with your Entrust digital ID (EPF file) and password.

If you need to administer more than one CA from a single Security Manager Administration installation, you must set entries in the `[Entrust ini override]` section of the `entrust.ini` file. For details, see [“Using Security Manager Administration with multiple Security Managers” on page 44](#).

By default, Security Manager Administration times out after the time (in minutes) specified by the **Login timeout (minutes)** policy attribute in your role's user policy. This prevents unauthorized access if you leave your computer unattended before logging out. When a timeout occurs, Security Manager locks and a dialog box appears that allows you to exit Security Manager Administration or log back in.

To increase or decrease the amount of time that must elapse before Security Manager Administration times out, configure your role's user policy as described in [“Modifying user policies” on page 402](#). Changing a user policy affects all roles that use that policy.

You can also manually lock Security Manager Administration (see [“Locking Security Manager Administration” on page 49](#)). If you lock Security Manager Administration, you can also log back in.

---

**Note:** Before you log in, make sure your computer clock is set correctly. If it is not set correctly, an error message appears when you try to log in. To solve this problem, synchronize the time on the computers hosting Security Manager Administration and Security Manager

---

This section contains the following procedures:

- [“To log in to Security Manager Administration” on page 47](#)
- [“To unlock Security Manager and log back in” on page 47](#)

## To log in to Security Manager Administration

- 1 Select **Start > All Programs > Entrust > Security Manager Administration**.

The **Entrust Authority (TM) Security Manager Administration Log in** dialog box appears.



- 2 Select your Entrust digital ID (EPF file) from the **User profile** drop-down list.  
If your profile does not appear in the drop-down list, click **Find Profile** to select your digital ID.
- 3 In the **Password** field, enter your password.
- 4 Click **OK**.

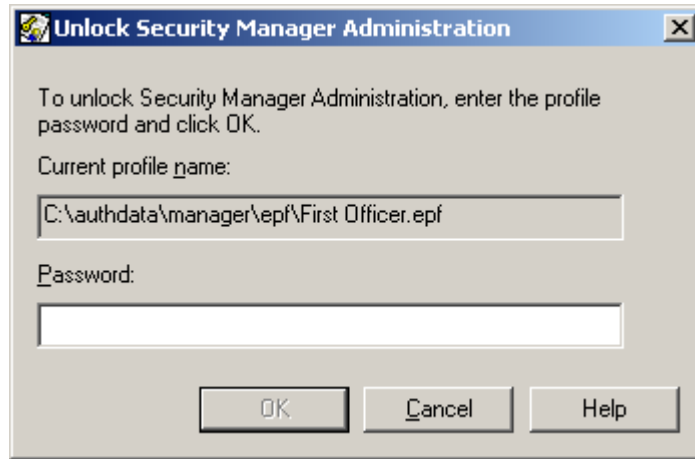
## To unlock Security Manager and log back in

- 1 When you lock Security Manager Administration or when Security Manager Administration times out, a dialog box similar to the following appears.



- 2 Click **OK**.

The **Unlock Security Manager Administration** dialog box appears.



- 3** In the **Password** field, enter your password.

After logging back in after a timeout, you are returned to where you were working when the timeout occurred.



# Locking Security Manager Administration

Lock Security Manager Administration if you plan on leaving your workstation unattended and you do not want to exit Security Manager Administration.

To manually lock Security Manager Administration, select **File > Lock**. The following dialog box appears.



To log back in, see ["Logging in to Security Manager Administration" on page 46](#).

# Showing and hiding the Security Manager Administration toolbar

In Security Manager Administration, you can show or hide the Security Manager Administration toolbar. The toolbar includes buttons that you can use to perform many common tasks, such as finding and adding users, performing audit log searches, and accessing the online help.

## To show and hide the Security Manager Administration toolbar

- 1 Log in to Security Manager Administration using your administrative profile. See [“Logging in to Security Manager Administration” on page 46](#).

Security Manager Administration appears.

- 2 Select **View > Toolbar**.

A check mark appears beside the Toolbar menu item when the toolbar is visible. Toggle the check mark to show and hide the toolbar.

You have now shown or hidden the Security Manager Administration toolbar.

# Changing your password

It is recommend that you change your password regularly. Changing your password regularly keeps your password secure and reduces the chance of a successful attack. Change your password if you even suspect that someone compromised your password.

## To change your password

- 1 If you are a Directory Administrator, log in to the Directory Browser (see [“Logging in to the Directory Browser” on page 70.](#)) Otherwise, log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46.](#))
- 2 Select **Operations > Change Password.**  
The **Change Profile Password** dialog box appears.

The image shows a Windows-style dialog box titled "Change Profile Password". The title bar is blue with a small icon on the left and a close button (X) on the right. The main area of the dialog is light gray. At the top, it says "Change the Entrust user password for: First Officer.epf". Below this, there are three text input fields. The first is labeled "Old password:". The second and third are grouped under a label "New password:" and are labeled "Password:" and "Confirm:" respectively. At the bottom of the dialog, there are four buttons: "OK", "Cancel", "Help", and "Password Rules >>>".

- 3 In the **Old password** field, enter your current password.
- 4 In the **Password** and Confirm fields, enter your new password.  
To view the password rules, click **Password Rules**. Your new password must conform to these rules.
- 5 Click **OK**.  
A dialog box appears confirming that the password change is successful.

# Authorizing sensitive operations

Roles include a setting that controls how many authorizations are required for sensitive operations. Most permissions allow specify whether the operations allowed by those permissions are sensitive operations that require authorization. (For more information about roles, see [“Administering roles” on page 353.](#))

If your role requires authorization for sensitive operations, one or more Entrust PKI administrators with sufficient permissions must authorize the operation. If your role requires one authorization, you can authorize the information yourself by providing your password. If your role requires more than one authorization, other Entrust PKI administrators must provide their profile and password to authorize the operation.

Entrust PKI administrators can store their digital ID (EPF file) on a token or in software. However, Security Manager Administration supports only two token readers and tokens must remain in the token readers until the operation completes. Therefore, a maximum of two token digital IDs can authorize operations and any other required authorizations must be performed by digital IDs stored in software.

---

**Note:** If you receive a “11530 Client was expecting DN change” error when authorizing a request, ensure that you have set up the permissions to the directory correctly. For more troubleshooting information for this error, see the Entrust TrustedCare Web site.

---

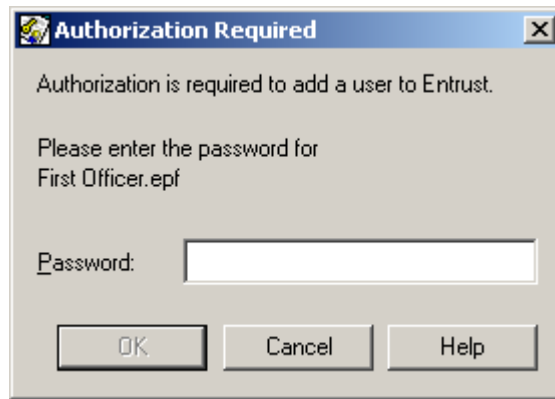
If you do not have enough administrators to authorize operations, a Security Officer must modify your role to decrease the number of authorizations required. If you do not have enough Security Officers, a Master User must decrease the number of Security Officer authorizations, or set Security Officers to key-recovery mode in Security Manager Control Command Shell (see the *Security Manager Operations Guide* for details).

This section contains the following procedures:

- [“To authorize a sensitive operation that requires one authorization” on page 53](#)
- [“To authorize a sensitive operation that requires multiple authorizations” on page 53](#)

## To authorize a sensitive operation that requires one authorization

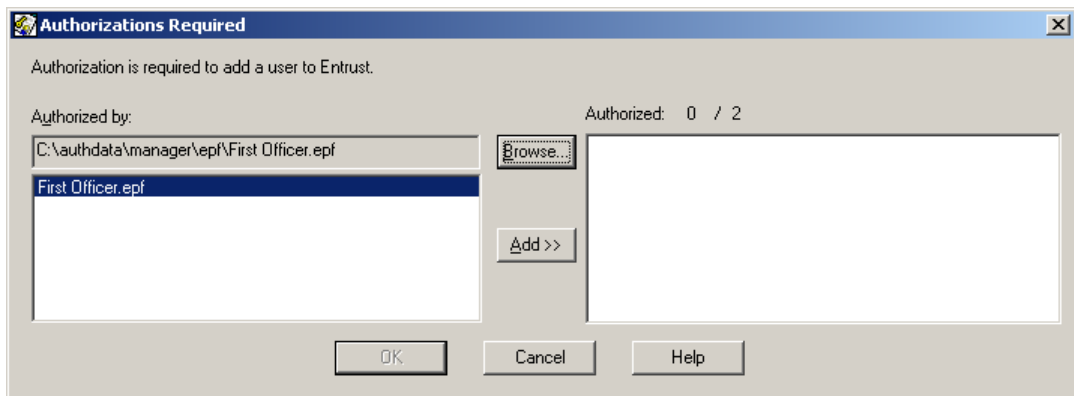
- 1 When you attempt to perform a sensitive operation that requires only one authorization, the **Authorization Required** dialog box appears.



- 2 Enter your password in the **Password** field and then click **OK**.

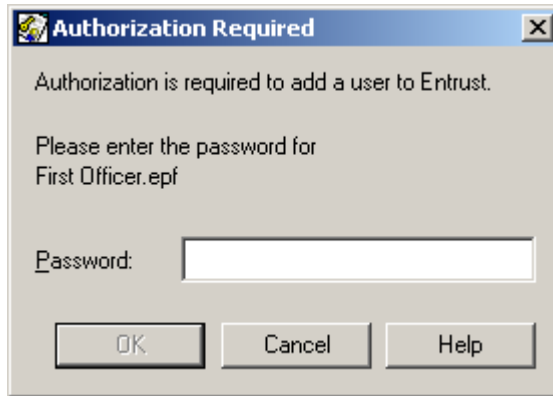
## To authorize a sensitive operation that requires multiple authorizations

- 1 When you attempt to perform a sensitive operation that requires more than one authorization, the **Authorizations Required** dialog box appears.



The dialog box lists up to 99 user profiles, allowing you to select the authorizations you need for the operation.

- 2 To add an authorization:
  - a Select a digital ID from the **Authorized by** list and then click **Add**, or click **Browse** to locate a profile.  
The **Authorization Required** dialog box appears.



- b** Enter the password for the selected profile and then click **OK**.  
The profile appears in the **Authorized** list. If the profile is on a token, do not remove the token until the operation completes. If you remove the token from the reader, Security Manager Administration displays a warning and ends the session.
- 3** Repeat the previous step until you acquire enough authorizations to complete the operation.
- 4** Click **OK**.

# Configuring the Security Manager license information

If you need to increase your Security Manager Enterprise or Web license, you must first purchase additional licenses from your Entrust sales representative. You will receive a new license card with updated license information. This section describes how to enter the new license information.

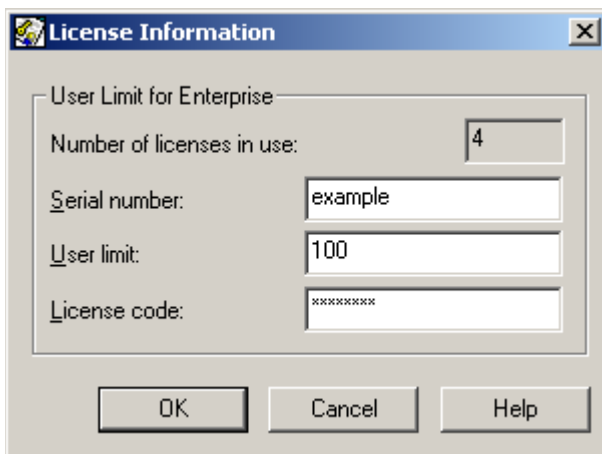
This section contains the following procedures:

- [“To change the Enterprise license information” on page 55](#)
- [“To change the Web license information” on page 56](#)

## To change the Enterprise license information

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Select **Operations > License Information > Enterprise**.

The **License Information** dialog box appears showing the number of licenses currently in use and your present license information.

A screenshot of the 'License Information' dialog box. The dialog has a title bar with a close button. Inside, there's a section titled 'User Limit for Enterprise'. Below this, there are four input fields: 'Number of licenses in use:' with the value '4', 'Serial number:' with the value 'example', 'User limit:' with the value '100', and 'License code:' with the value 'xxxxxxx'. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

- 3 Enter the information from your new license card in the **Serial number**, **User limit** and **License code** fields and then click **OK**.
- 4 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

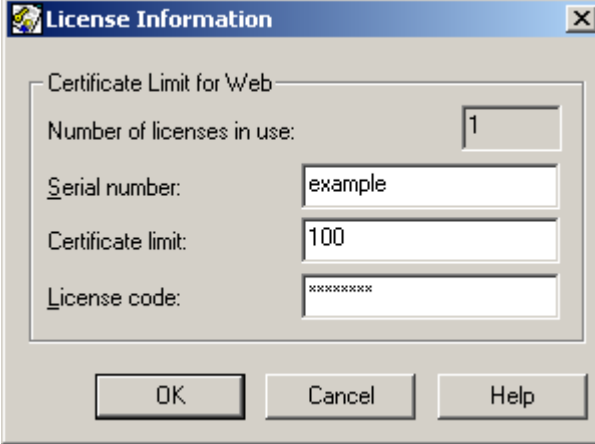
If the operation was successful, a success message appears.

## To change the Web license information

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).

- 2 Select **Operations > License Information > Web**.

The **License Information** dialog box appears showing the number of licenses currently in use and your present license information.

The image shows a Windows-style dialog box titled "License Information". It has a standard title bar with a minimize button, a maximize button, and a close button. The main content area is divided into two sections. The top section is titled "Certificate Limit for Web" and contains a label "Number of licenses in use:" followed by a text box containing the number "1". The bottom section contains three labels with corresponding text boxes: "Serial number:" with the text "example", "Certificate limit:" with the text "100", and "License code:" with the text "xxxxxxx". At the bottom of the dialog box are three buttons: "OK", "Cancel", and "Help".

- 3 Enter the information from your new license card in the **Serial number**, **Certificate limit** and **License code** fields and then click **OK**.

- 4 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.



# Configuring Security Manager Administration preferences

Preferences determine how Security Manager Administration behaves for certain operations such as directory searches. Configuring preferences can improve the performance of Security Manager Administration.

This section contains the following topics:

- [“Configuring general preferences” on page 57](#)
- [“Configuring directory search preferences” on page 59](#)
- [“Configuring search performance preferences” on page 60](#)
- [“Configuring user list preferences” on page 62](#)
- [“Configuring group information preferences” on page 64](#)

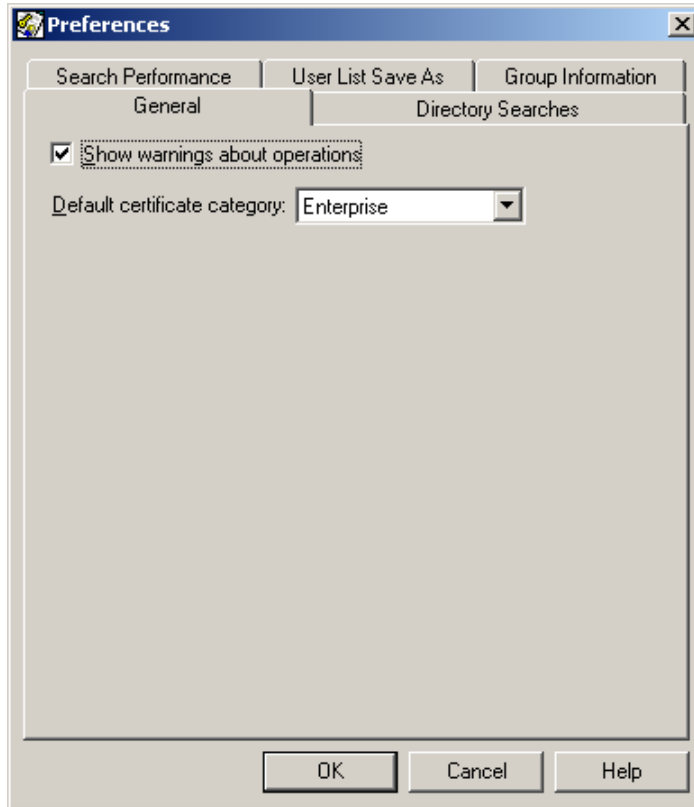
## Configuring general preferences

General preferences control whether warnings appear when you perform operations in Security Manager Administration, and specifies the default certificate category for certificate category drop-down lists.

### To configure general preferences

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Select **File > Preferences**.

The **Preferences** dialog box appears



- 3** Click the **General** tab.
- 4** To show warnings when performing operations, select **Show warnings about operations**. Otherwise, deselect the option.

---

**Note:** It is strongly recommended that you select **Show warnings about operations** unless you are very familiar with Security Manager operations.

---

- 5** In the **Default certificate category** drop-down list, select the default certificate category that appears in certificate category drop-down lists.  
If you do not have a Web license, **Enterprise** is the only certificate category in the list.
- 6** Click **OK**.
- 7** If prompted to authorize the operation, authorize the operation. See ["Authorizing sensitive operations" on page 52](#).  
If the operation was successful, a success message appears.

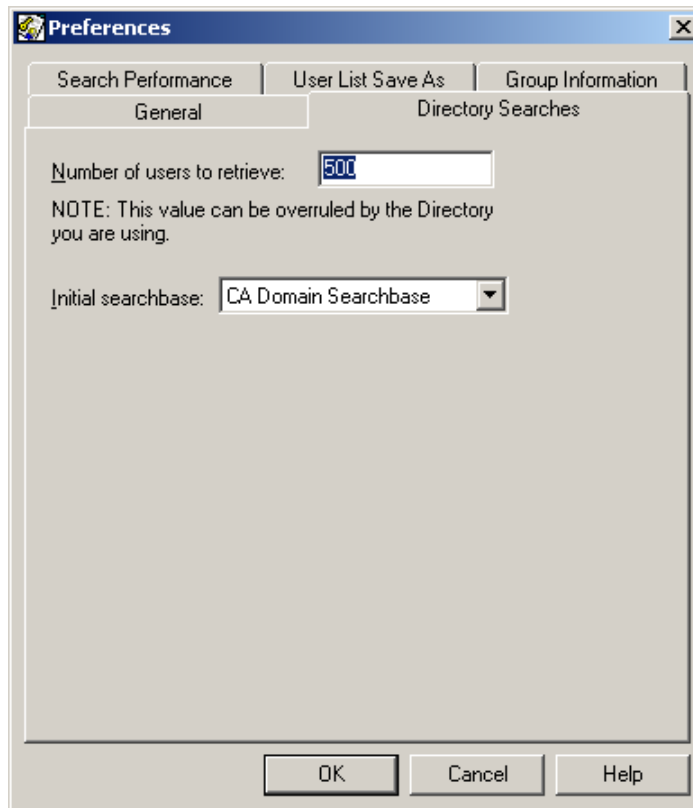
## Configuring directory search preferences

Directory search preferences control how many users to retrieve during directory searches, and specifies the default searchbase.

### To configure directory search preferences

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Select **File > Preferences**.

The **Preferences** dialog box appears.



- 3 Click the **Directory Searches** tab.
- 4 In the **Number of users to retrieve** field, enter the number of users (from 1 to 500) to return in directory searches.

---

**Note:** The number you specify here may be overruled by the directory you are using if it has a lower retrieval limit.

---

The larger the number you enter, the longer the wait to see search results if you select broad search parameters.

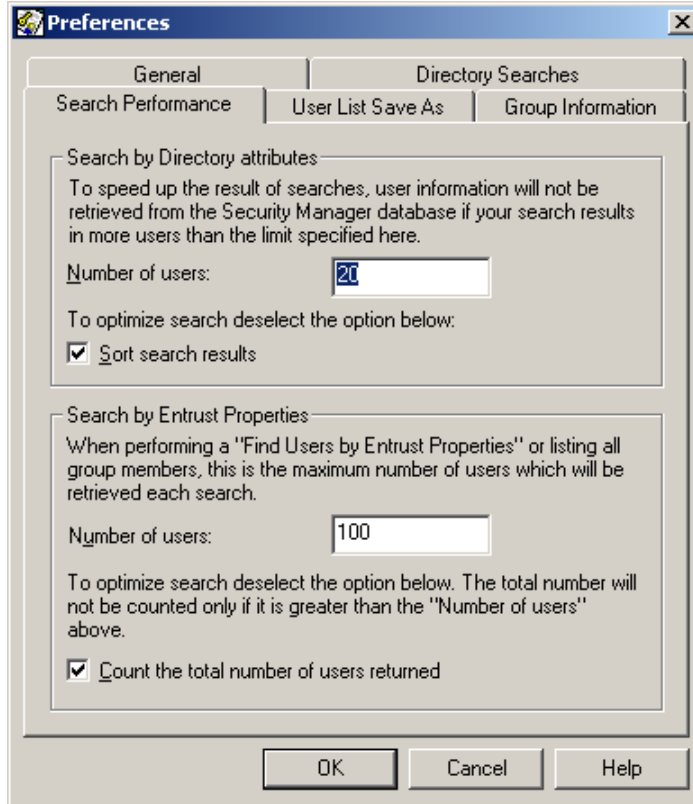
- 5** In the **Initial searchbase** drop-down list, select the default searchbase for directory searches.
- 6** Click **OK**.
- 7** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
If the operation was successful, a success message appears.

## Configuring search performance preferences

Search performance preferences control how Security Manager Administration searches the Security Manager database. Configuring these preferences can improve the performance of database searches. See the *Entrust Authority Security Manager Operations Guide* for ways to control user searching in large databases.

### To configure search performance preferences

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Select **File > Preferences**.  
The **Preferences** dialog box appears.



- 3 Click the **Search Performance** tab.
- 4 The options under **Search by Directory attributes** control how Security Manager Administration searches the database when you search by directory attributes (see [“Finding users by directory attributes” on page 142](#)):
  - In the **Number of users** field, enter the maximum number of users for which you want to immediately retrieve database information.

Searching the Security Manager database takes time. If your search finds more users than the number specified, Security Manager Administration does not retrieve user information. User information is retrieved for a specific user when you select a user from the search results.
  - To sort the search results in alphabetical order (by distinguished name), select **Sort search results**. To display the search results in the order that they appear in the database, deselect the option.
- 5 The options under **Search by Entrust Properties** control how Security Manager Administration searches the database when you search by Entrust properties (see [“Finding users by Entrust properties” on page 134](#)):

- In the **Number of users** field in the **Search by Entrust Properties** pane, enter the number of users returned for each search.

If a search finds more users than the number specified, a dialog box appears asking if you want to continue searching.

- If a search finds more users than the value specified by **Number of users**, a dialog box appears asking if you want to continue searching. To include the total number of users that meet your search criteria, select **Count the total number of users returned**. To exclude the total number of users that meet your search criteria, deselect the option.

**6** Click **OK**.

**7** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

## Configuring user list preferences

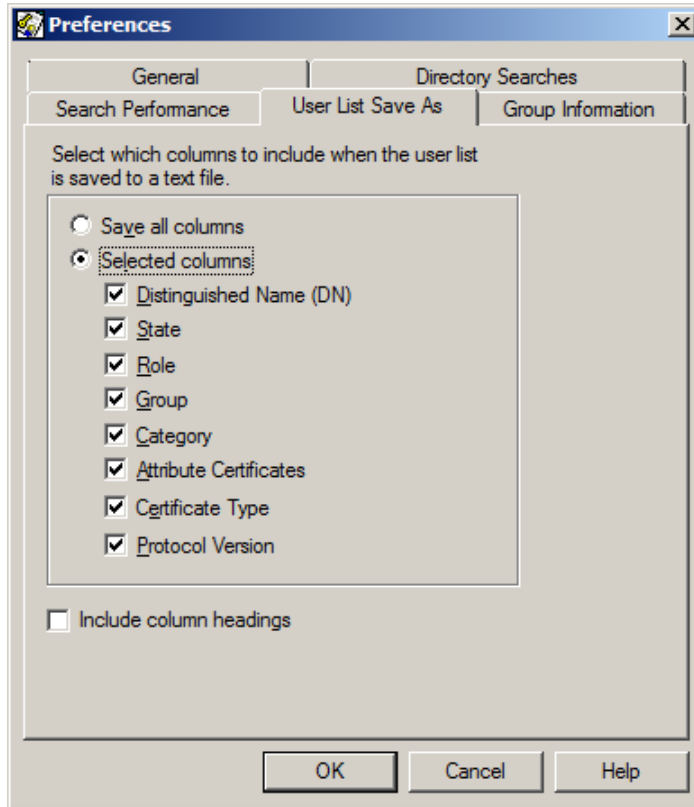
When saving a user list to a file (see [“Finding users” on page 134](#)), Security Manager Administration includes user information such as the user’s role and group. The user list preferences determine the user information that is included in the file.

### To configure user list preferences

**1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).

**2** Select **File > Preferences**.

The **Preferences** dialog box appears.



- 3 Click the **User List Save As** tab.  
The **User List Save As** property page appears.
- 4 To save all the user information to the file, click **Save all columns**. Otherwise, click **Selected columns** and then select the check boxes for each column of user information you want to include, and deselect the check boxes for each column of user information you want to exclude.
- 5 To include the column headings in the file, select **Include column headings**.  
Including the column headings along with the user list can make your text file easier to read.
- 6 Click **OK**.
- 7 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
If the operation was successful, a success message appears.

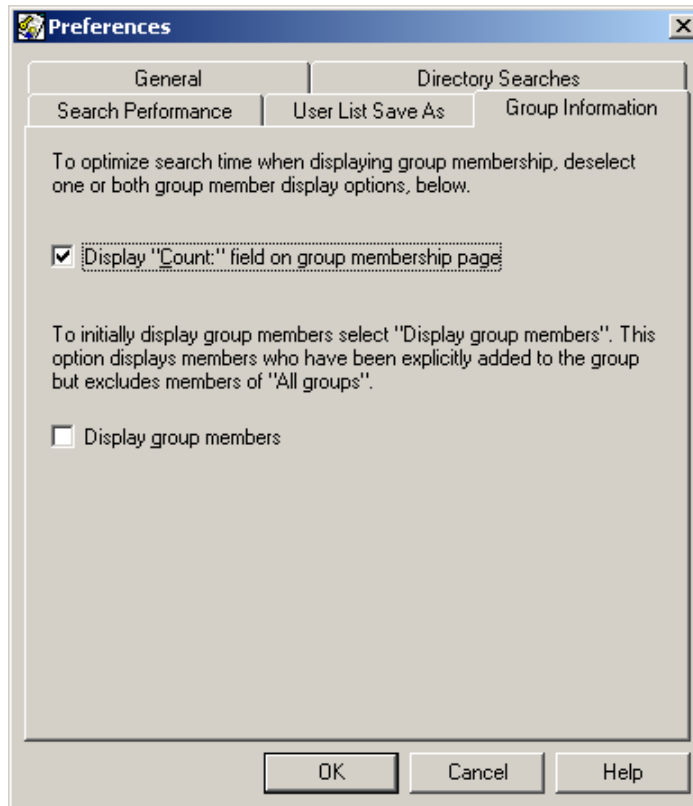
## Configuring group information preferences

Group information preferences control how Security Manager displays group membership information. When viewing groups (see [“Viewing groups” on page 330](#)), you may experience slower access times if you have large number of users in your groups. Configuring these preferences can improve the access times when viewing groups.

### To configure group information preferences

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Click **File > Preferences**.

The **Preferences** dialog box appears.



- 3 Click the **Group Information** tab.
- 4 To display the number of group members when viewing groups, select **Display “Count:” field on group membership page**.



- 5 To automatically display group members (except users who belong to all groups) when viewing groups, select **Display group members**.
- 6 If prompted to authorize the operation, authorize the operation. See ["Authorizing sensitive operations" on page 52](#).  
If the operation was successful, a success message appears.

# Recovering your profile

If you lose your profile or forget your password, you must recover your profile. Recovering a profile is a two-step process:

- 1 A PKI administrator sets you up for key recovery (see [“Recovering user key pairs” on page 162](#)). If you are a Security Officer, a Master User can set you up for key recovery (see the *Security Manager Operations Guide*).

Setting you up for key recovery generates a new reference number and authorization code.

- 2 Use the new reference number and authorization code to recover the profile.

Security Manager Administration is a V1 client application. You can recover your profile with Security Manager Administration if you are a V1 user. If you are a V2 user, you must use a V2 client application (such as Entrust Entelligence Security Provider for Windows) to recover your profile, or another PKI administrator can convert you to a V1 user (see [“Converting V2 users to V1 users” on page 247](#)).

## To recover your profile

- 1 Obtain the reference number and authorization code from an Entrust PKI administrator or Master User.
- 2 Select **Start > Programs > Entrust > Security Manager Administration**.

The **Entrust Authority (TM) Security Manager Administration Log In** dialog box appears.



- 3 Click **Recover Profile**.

The **Recover profile** dialog box appears.

**Recover profile**

Enter your profile name and location

Name:

Location:

Enter your activation codes

Reference number:

Authorization code:

Enter the password for your profile

Password:

Confirm:

- 4** In the **Name** field, enter the name of your profile. For example, if the name of your profile is `Example.epf`, enter `Example`.
- 5** In the **Location** field, enter the location where you store your profile, or click **Browse** to select the location.
- 6** In the **Reference number** field, enter the reference number you received the Entrust PKI administrator or Master User.
- 7** In the **Authorization code** field, enter the authorization code you received the Entrust PKI administrator or Master User.
- 8** In the **Password** and **Confirm** fields, enter a new password for your profile.
- 9** Click **OK**.
- 10** A message appears stating that your profile already exists. The message also asks if you want to overwrite your existing profile. Click **Yes** to recover your profile.



## Using the Directory Browser

Entrust PKI administrators with sufficient permissions (such as users with the Directory Administrator role) can use the Directory Browser to perform directory-related tasks, such as adding and deleting directory entries.

Entrust PKI administrators with sufficient permissions can add, change, and delete directory attributes using the Directory Browser. Directory attributes are not absolute and vary depending the Security Manager deployment of your organization. You may need certain information before you can add, change, or delete an attribute. If you are unsure which attributes you can safely add, change, or delete, contact an administrator with advanced knowledge of your directory.

If you use Microsoft Active Directory, you must use your Active Directory tools to perform directory-related tasks. You cannot use the Directory Browser to administer Microsoft Active Directory.

This chapter contains the following sections:

- [“Logging in to the Directory Browser” on page 70](#)
- [“Locking the Directory Browser” on page 71](#)
- [“Finding entries in the directory” on page 72](#)
- [“Adding entries to the directory” on page 74](#)
- [“Deleting directory entries” on page 76](#)
- [“Adding attributes to directory entries” on page 77](#)
- [“Adding directory attribute values” on page 79](#)
- [“Replacing directory attribute values” on page 80](#)
- [“Deleting directory attributes or attribute values” on page 81](#)
- [“Changing the Security Manager Directory Administrator password” on page 82](#)

# Logging in to the Directory Browser

This section describes how to log in to the Directory Browser to administer the Security Manager directory.

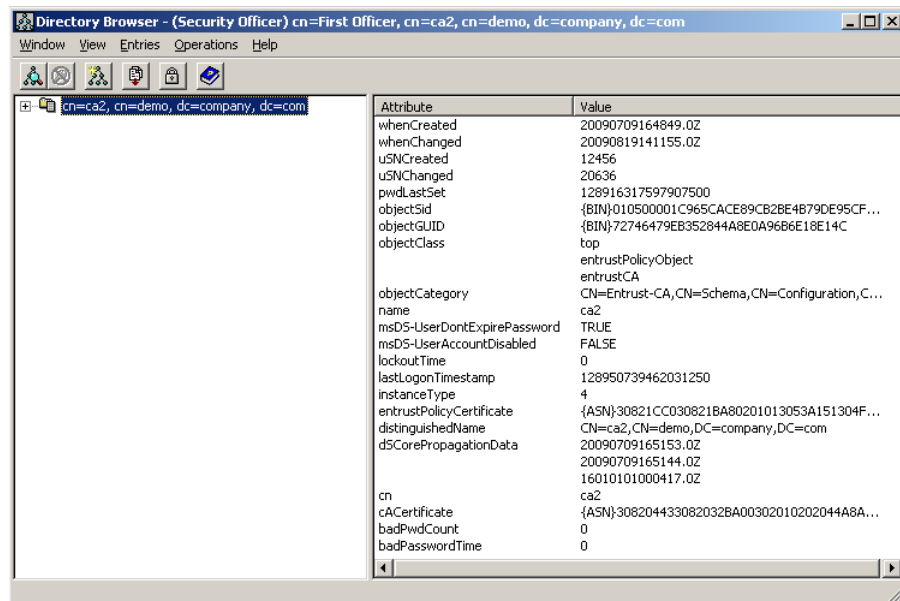
## To log in to the Directory Browser

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration”](#) on page 46).

For Entrust PKI administrators that can only use the Directory Browser (such as users with the Directory Administrator role), the Directory Browser window appears. Other Entrust PKI Administrators see the main Security Manager Administration window.

- 2 If the main Security Manager Administration window appears, select **Operations > Directory Browser**.

The Directory Browser window appears.



# Locking the Directory Browser

Lock the Directory Browser if you plan on leaving your workstation unattended and you do not want to exit the Directory Browser.

To manually lock the Directory Browser, select **Window > Lock**. The following dialog box appears.



To log back in, see ["Logging in to Security Manager Administration" on page 46](#).

# Finding entries in the directory

Use the following procedure to find entries in the directory. To find users and entries using Security Manager Administration, see [“Finding users” on page 134](#).

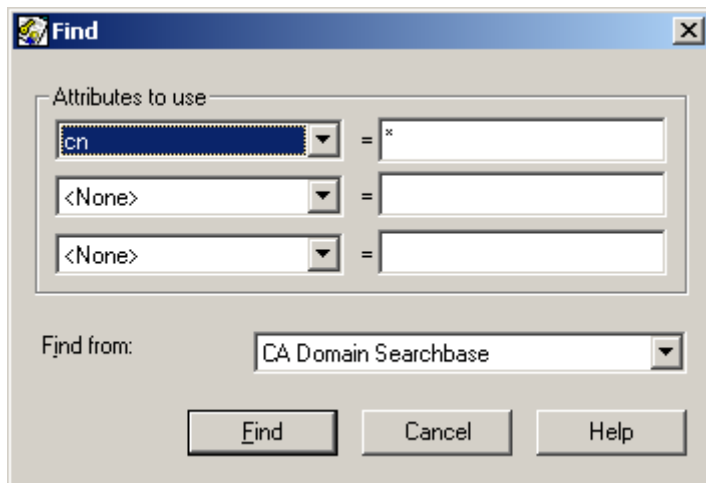
To control the number of entries returned in searches, see [“Configuring directory search preferences” on page 59](#).

## To find a directory entry

- 1 Log in to the Directory Browser. See [“Logging in to the Directory Browser” on page 70](#).

- 2 Select **Entries > Find**.

The **Find** dialog box appears.



- 3 Select directory attributes in the **Attributes to use** drop-down lists.

The list of directory attributes in the drop-down lists is configured in your `entrust.ra.ini` file. See the *Security Manager Operations Guide* for details.

- 4 For each attribute you selected, enter the attribute value you want to search for in the **Attributes to use** field.

When you search for entries with directory attribute values that include special characters, the values you enter must match the directory entry.

To decrease the amount of time it takes to find entries, specify narrow criteria for your search.

Optionally, you can use wildcards. Wildcards let you search for partial attributes. Add an asterisk (\*) with a partial search string to find users that include the search string information. For example, enter `*dr*` to find all entries named `Andrew` and



Drew. Enter `dr*` to exclude Andrew and find only entries beginning with the letters `dr`.

- 5 Choose a searchbase in the **Find from** drop-down list to find entries in that searchbase. For more information about searchbases, see [“Administering searchbases” on page 341](#).
- 6 Click **Find**.

---

**Note:** If you select **Entries > Cancel Find** or click the **Cancel** button in the toolbar before the search is complete, only the entries that are found up to that point are displayed. Entries are not found in any particular order. Even if you notice results that range from A-Z after canceling the search, it does not mean that you have found every entry in the directory that meets your search criteria.

---

The entries matching your search criteria appear in the tree view in the Directory Browser.

# Adding entries to the directory

Entrust PKI administrators with sufficient permissions can add entries to the directory. You might add a new directory entry when a new employee joins your organization.

## To add an entry to the directory

- 1 Log in to the Directory Browser. See [“Logging in to the Directory Browser” on page 70.](#)
- 2 Select **Entries > New Entry**.

The **New Entry** dialog box appears. An asterisk (\*) appears beside required fields. You must enter information into all fields marked with asterisks.

The information required under the **Naming** tab may differ from that described in this procedure if a Security Officer or Entrust PKI administrator has modified the template definition file. Contact a Security Officer if you have any questions about the appearance of this dialog box.

The screenshot shows the 'New Entry' dialog box with the 'Naming' tab selected. The 'Type' dropdown is set to 'Person'. Below it is a text box containing 'This is the user type to be used for most Entrust users.' The 'In DN' section contains four fields: 'First Name' (required), 'Last Name' (required), 'Serial Number', and 'Email'. The 'First Name' and 'Last Name' fields have checkboxes in the 'In DN' column, which are checked. The 'Serial Number' and 'Email' fields have unchecked checkboxes. Below the fields is a note: 'Asterisks (\*) appear beside required attributes.' The 'Add to:' dropdown is set to 'CA Domain Searchbase'. There is a 'Show DN...' button. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Attribute	In DN
* First Name	<input checked="" type="checkbox"/>
* Last Name	<input checked="" type="checkbox"/>
Serial Number	<input type="checkbox"/>
Email	<input type="checkbox"/>

- 3** In the **Type** drop-down list, select an entry type (such as **Person** or **Web server**).  
You can also select **Organizational Unit** to create an **ou=** entry in the directory.  
You can only create organizational unit entries using the Directory Browser.  
Typically, you create organizational units to create a new searchbase. See [“Adding searchbases” on page 344](#) for more information.
- 4** If you selected **Person**:
  - In the **First Name** field, enter the user’s first name.
  - In the **Last Name** field, enter the user’s last name.
  - (Optional.) In the **Serial Number** field, enter the user’s serial number (for example, the employee number).  
Depending on your organization, the serial number may not be the employee number. Check with a Security Officer if you are unsure which number to enter.
  - (Optional.) In the **Email** field, enter the user’s email address. To enter more than one email address, separate each email address multiple with a space. For more information, see [“Email addresses in Security Manager” on page 132](#).
- 5** If you selected **Web server**:
  - In the **Name** field, enter a name for the entry.
  - (Optional.) In the **Description** field, enter a description for the entry.
- 6** If you selected **Organizational Unit**, enter a name for the entry in the **Organization Unit** field.
- 7** In the **Add to** drop-down list, select the searchbase where you want to add the user.  
For more information about searchbases, see [“Administering searchbases” on page 341](#).
- 8** Click **OK**.

# Deleting directory entries

Entrust PKI administrators with sufficient permissions can delete entries from the directory. Delete directory entries when you created a new entry by mistake or if you no longer need the directory entry. For example, when an employee leaves your organization, you can delete the employee's corresponding directory entry.

When you delete a directory entry, the corresponding user is not deleted from the Security Manager database. Before deleting a directory entry, you should deactivate, revoke, or archive the user.

## To delete a directory entry

- 1** Log in to the Directory Browser (see [“Logging in to the Directory Browser” on page 70](#)).
- 2** If you do not see the entry that you want to delete in the tree view, search for the entry (see [“Finding entries in the directory” on page 72](#)).
- 3** Select the entry that you want to delete and then select **Entries > Selected Entry > Delete**.
- 4** If a confirmation dialog box appears, click **OK** if you are sure that an Administrator has already deactivated the user. Contact an Administrator if you are unsure.

# Adding attributes to directory entries

You can add an attribute to a directory entry for any of the following reasons:

- To include missing information.  
It may happen that the template definition file is changed to include or exclude an attribute.
- To re-add attributes that are accidentally deleted.

While you can enter any data in the name and value fields in the **Add Attribute** dialog box, the attributes permitted by your directory schema vary. Consult an administrator with advanced knowledge of your directory if you have any questions about which attributes are valid and which attributes are invalid.

You can also add attributes to directory entries in bulk. This is useful when adding a known value to all entries. For example, your organization may need to include employee phone numbers in its directory entry information. While you can add this information to each entry individually, it is more efficient to do it in bulk. See ["Adding and deleting directory attributes in bulk" on page 323](#).

---

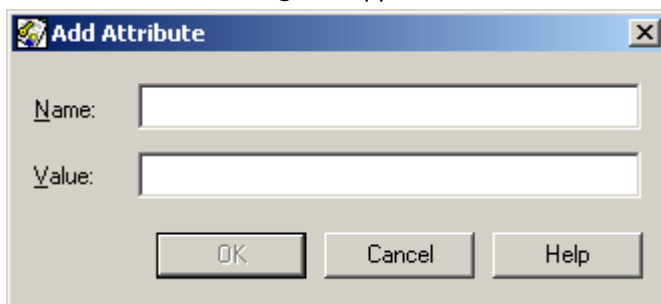
**Note:** When you add, change, or delete an attribute in the DN, you may have to make the same change to the user in the Security Manager database. See ["Modifying distinguished names" on page 192](#).

---

## To add an attribute to a directory entry

- 1 Log in to the Directory Browser (see ["Logging in to the Directory Browser" on page 70](#)).
- 2 If you do not see the entry that you want to modify in the tree view, search for the entry (see ["Finding entries in the directory" on page 72](#)).
- 3 Select the directory entry you want to modify and then select click **Entries > Selected Entry > Add Attribute** in the pop-up menu.

The **Add Attribute** dialog box appears.



- 4** In the **Name** field, enter the name of the attribute.
- 5** In the Value field, enter the value of the attribute.
- 6** Click **OK**.

Error checking prevents you from adding directory attributes that are not supported in the directory. You receive an error message when you enter an attribute that your directory does not support.

# Adding directory attribute values

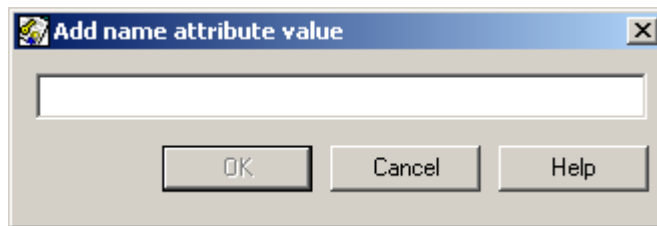
You can add additional values to an existing directory attribute.

You should be familiar with the conventions of your organization's directory before you add an attribute value to a directory attribute. Consult an administrator with advanced knowledge of your directory if you have any questions about which attribute values are valid and which are not.

## To add a directory attribute value

- 1 Log in to the Directory Browser (see [“Logging in to the Directory Browser” on page 70](#)).
- 2 In the tree view, select a directory entry.
- 3 In the right pane, right-click an attribute value, and then select **Add**.

A dialog box appears in which you can add a directory attribute value. This example shows one of the dialog boxes that can appear.



- 4 Enter the attribute value and click **OK**.

Error checking prevents you from adding directory attributes that are not supported in the directory. You receive an error message when you enter an attribute that your directory does not support.

# Replacing directory attribute values

Before you replace a directory attribute value, make sure that the attribute value is one that you can safely replace. You should not replace attribute values that comprise a DN or that are required by your security policy.

---

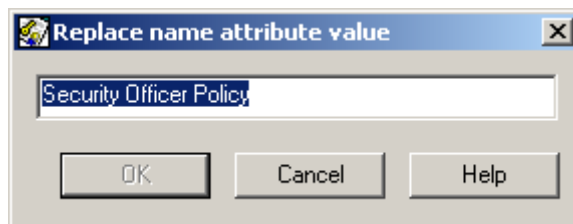
**Note:** An error message appears when you try to change to an attribute value that your directory does not support.

---

## To replace a directory attribute value

- 1 Log in to the Directory Browser (see [“Logging in to the Directory Browser”](#) on page 70).
- 2 In the tree view, select a directory entry.
- 3 In the right pane, right-click an attribute value, and select **Replace**.

A dialog box appears in which you can replace directory information. This example shows one of the dialog boxes that can appear.



- 4 Enter the new information and then click **OK**.



# Deleting directory attributes or attribute values

Before you delete an attribute value, make sure that the attribute value is one that you can safely delete. You should not delete vital attribute values that comprise a DN or that are required by your security policy.

---

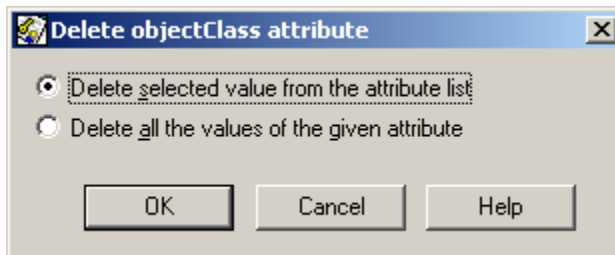
**Note:** When you add, change, or delete an attribute in the DN, you may have to make the same change to the user in the Security Manager database. See [“Modifying distinguished names” on page 192](#).

---

## To delete a directory attribute or attribute value

- 1 Log in to the Directory Browser (see [“Logging in to the Directory Browser” on page 70](#)).
- 2 In the tree view, select a directory entry.
- 3 In the right pane, right-click a directory attribute value and select **Delete**.

If the directory attribute has multiple values, a dialog box appears asking you if you want to delete just the selected value or all of the values. To delete a single directory attribute value, choose the first option and click **OK**.



If a directory attribute has more than one value and you delete only one, only that value is deleted. If you delete all values, you also delete the directory attribute.

# Changing the Security Manager Directory Administrator password

Security Manager Administration uses the Security Manager Directory Administrator password to authenticate itself to the directory to perform such tasks as adding and deleting users.


Entrust PKI Administrators with sufficient permissions can change the Directory Administrator password.

It is recommend that you change the Directory Administrator password regularly. Changing the password regularly keeps the password secure and reduces the chance of a successful attack. Change the password if you even suspect that someone compromised the password.

## To change the Entrust Directory Administrator password in Entrust

- 1 Change the Entrust Directory Administrator password in the directory, using your directory tools. Consult the documentation for your directory for details.
- 2 Log in to the Directory Browser. See [“Logging in to the Directory Browser” on page 70](#).
- 3 Select **Operations > Change Directory Password**.

The **Change Directory Password** dialog box appears.

A screenshot of the 'Change Directory Password' dialog box. The title bar is blue with a yellow icon on the left and a close button on the right. The main area has a light gray background. At the top, it says 'Change the Directory Access password for the Directory.' Below this is a section titled 'New password' which contains two text input fields: 'Password:' and 'Confirm:'. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

- 4 Enter the new password in the **Password** and **Confirm** field, and then click **OK**. A dialog box appears to confirm the password change.
- 5 Click **OK**.

## Setting up the Certification Authority

This chapter describes how to configure your Certification Authority (CA) to achieve the specific security goals of your organization. For example, after installing Security Manager, you need to determine settings in the Security Policy section of Security Manager Administration. You may also want to create searchbases or administrators.

Read this chapter if you are a Security Officer or if you have just finished installing Security Manager.

To configure some of the settings described in this chapter, you require adequate permissions (such as those of a Security Officer). For a description of roles and administrative permissions, see ["Administering roles" on page 353](#).

This chapter includes the following sections:

- ["Configuring the Security Policy" on page 84](#)
- ["Exporting the current CA certificate" on page 102](#)
- ["Viewing CA information" on page 104](#)

# Configuring the Security Policy

Before adding administrators and other users to Security Manager, you should review, and, if necessary, customize settings in the following categories under **Security Policy** in Security Manager Administration:

- **General Security Policy properties.** These include the Administration Policy, Encryption and Verification OIDs (object identifiers), and Queued Requests.
- **Properties of certificate categories.** These include the option to put default policy OIDs in certificates, to allow for unknown extensions, and to set default key lifetimes.
- **User Policies.** These specify a user's default certificate settings according to the user's role. User policies include such settings as certificate contents, password rules, and algorithms used.
- **Roles.** These define who does what to administer Security Manager. For more information about roles, see [“Administering roles” on page 353](#).

In addition to reviewing the Security Policy settings in Security Manager Administration, you should also review your organization's certificate practice statement, and update it if necessary.

A certificate practice statement is a document you create that describes, in detail, the policies and procedures that Entrust PKI administrators must follow when operating your PKI. It is an extension of your Security Policy settings. For example, your certificate practice statement can specify such things as where and how you store backup tapes of your Security Manager database, and what credentials a user must present before they become an Entrust PKI administrator for your PKI.

Ideally, a certificate practice statement should address all the security-related issues in the operation of your PKI that the Security Manager software cannot control directly.

The following sections describe how to determine these settings:

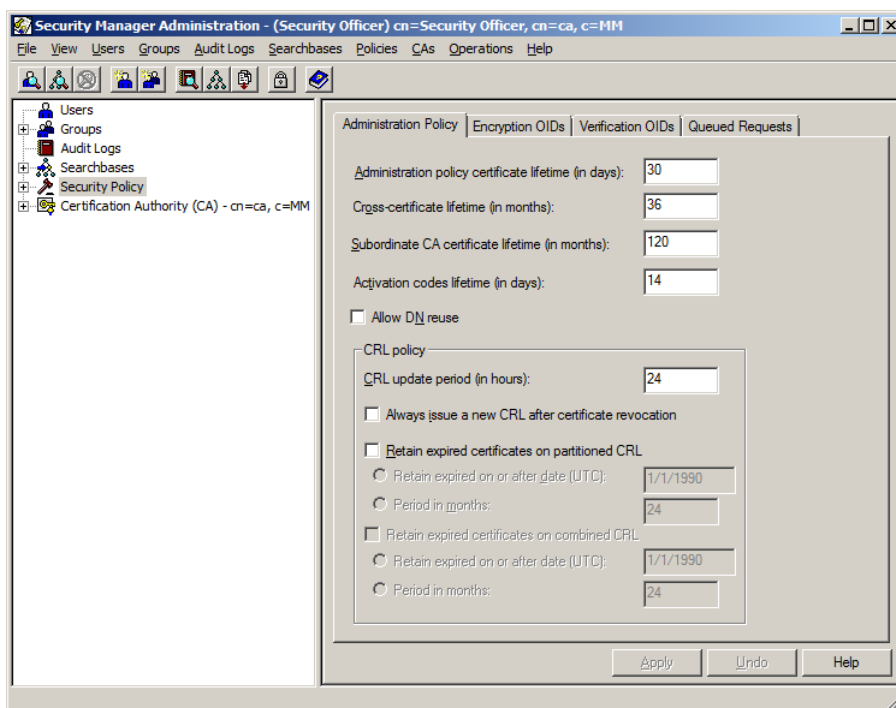
- [“Configuring the Administration Policy” on page 85](#)
- [“Interaction between CRL settings in the Security Policy” on page 88](#)
- [“Configuring the encryption and verification OIDs” on page 90](#)
- [“Configuring queued requests” on page 95](#)
- [“Configuring options for the various certificate categories” on page 97](#)

## Configuring the Administration Policy

The administration policy includes various settings such as certificate lifetimes, the Certificate Revocation List (CRL) policy, and the key pair algorithm and size. The following steps describe each setting in detail. The section [“Interaction between CRL settings in the Security Policy” on page 88](#) describes an interaction between one of the Administration policy settings (**Always issue a new CRL after certificate revocation**) and an administrative permission (**Force CRLs**).

### To set the Administration Policy

- 1 Log in to Security Manager Administration as a Security Officer (see [“Logging in to Security Manager Administration” on page 46](#)).



- 2 In the tree view, click **Security Policy**.
- 3 Click the **Administration Policy** tab.
- 4 In the **Administration policy certificate lifetime (in days)** field, enter a number from 1 to 3650, or accept the default (30 days).

---

**Note:** Policy certificates (including the Administration policy certificate) are reissued every 24 hours. Each new certificate has the full lifetime you specify in this dialog box.

---

- 5 In the **Cross-certificate lifetime (in months)** field, enter a number from 2 to 420, or accept the default (36 months). See [“Cross-certifying with other CAs” on page 465](#) for more information.
- 6 In the **Subordinate CA certificate lifetime (in months)** field, enter a number from 2 to 420, or accept the default (120 months).
- 7 In the **Activation codes lifetime (in days)** field, enter a number from 1 to 365, or accept the default (14 days). This setting controls the lifetime of user activation codes only. It does not control the lifetime of activation codes for subordinate CAs, which is fixed at three days.
- 8 Select **Allow DN reuse** to allow new users to reuse any previously used DN. If deselected, new users cannot reuse a previously used DN.

For example, consider a scenario where a user changes DNs from `cn=Jane Smith,c=CA` to `cn=Jane Brown,c=CA`.

- If **Allow DN reuse** is selected, another user can use `cn=Jane Smith,c=CA`.
- If **Allow DN reuse** is deselected, only user1 can use `cn=Jane Smith,c=CA`.

Regardless of the **Allow DN reuse** setting, users can always reuse a DN they have previously used, as long as it is not currently used by another user.

---

**Note:** It is recommended that you do not reuse the DN of an archived user, in case you need to retrieve the archived user in the future.

---

- 9 In the **CRL update period (in hours)** field, enter how often (in hours) that Security Manager automatically updates the combined CRL (if enabled) and the partitioned CRLs in the directory.

Enter a value from 4 to 48 hours. By default, Security Manager automatically updates the CRLs in the directory every 24 hours. Your certificate policy statement should specify the lifetimes of CRLs in your CA domain. Network and user performance affects how often you should update CRLs.

---

**Note:** If you defined custom CRL lifetimes in Security Manager Control Command Shell, their values override the **CRL update period** specified here. If you have any doubt about how to set the CRL lifetime, consult a Security Manager Master User.

---

- 10 Select **Always issue a new CRL after certificate revocation** if you want Security Manager to automatically update the applicable CRL immediately after a certificate is revoked.

This option interacts with the Entrust PKI administrator permission **Force CRLs**. For more information, see [“Interaction between CRL settings in the Security Policy” on page 88](#).

- 11** The **Retain expired certificates on partitioned CRL** option controls whether revoked certificates that have expired are retained on partitioned CRLs and ARLs.

To remove all expired certificates from partitioned CRLs and ARLs, deselect **Retain expired certificates on partitioned CRL**. If deselected, expired certificates are removed from a partitioned revocation lists when the following conditions are met:

- the certificate has been revoked for at least 72 hours
- the revocation list was published at least once after the certificate was revoked

To retain expired certificates on partitioned CRLs and ARLs, select **Retain expired certificates on partitioned CRL** and then select one of the following options:

- To retain all revoked certificates on partitioned revocation lists that expired after a specific date, select **Retain expired on or after date (UTC)**, and then enter the date in DD/MM/YYYY format. Do not include leading zeros.

For example, if you enter 1/1/2010 (January 1, 2010), certificates that expired on or after 1/1/2010 will remain on partitioned revocation lists, while certificates that expired before 1/1/2010 will be removed.

The date must be from 1/1/1990 (January 1, 1990) to 14 days before the current date at 00:00 UTC. For example, if today is 12/31/2020, (December 31, 2020), you must enter 12/17/2020 or earlier, or you will receive an error.

To retain all expired certificates on partitioned revocation lists, use the value 1/1/1990 (January 1, 1990). If the date is 1/1/1990, Security Manager skips expired revocation checks when updating revocation lists. Skipping expired revocation checks increases the performance of Security Manager.

- To retain expired certificates for a specific number of months, select **Period in months** and then enter how many months (from 1 to 240) expired certificates will remain on partitioned CRLs and ARLs.

For example, if you enter 24 months, certificates that have been expired for longer than 24 months will be removed from partitioned revocation lists.

---

**Note:** Changing the **Retain expired certificates on partitioned CRL** option will not take effect until Security Manager checks partitioned CRLs and ARLs for expired certificates. Security Manager will add expired certificates or remove expired certificates as required when it updates the partitioned revocation lists. A Master User can issue updated partitioned CRLs and ARLs immediately with an expired revocation check by running the `rl issue -crl -arl -checkexpire` command (see the *Security Manager Operations Guide*).

---

For more information about revoking certificates, see [“Revoking user certificates” on page 174](#).

**12** The **Retain expired certificates on combined CRL** option shows whether revoked certificates that have expired are retained on combined CRLs. The **Retain expired certificates on combined CRL** option is always disabled. Only a Master User can set this option in Security Manager by configuring the `ExpiredOnCombinedCRL` advanced setting. See the *Security Manager Operations Guide* for details.

If **Retain expired certificates on combined CRL** is deselected, expired certificates are removed from combined CRLs when the following conditions are met:

- the certificate has been revoked for at least 72 hours
- the CRL was published at least once after the certificate was revoked

If **Retain expired certificates on combined CRL** is selected:

- The **Retain expired on or after date (UTC)** option specifies a date (DD/MM/YYYY format). Revoked certificates that expired before this date are removed from combined CRLs.

For example, if the date specified is 1/1/2010 (January 1, 2010), certificates that expired on or after 1/1/2010 will remain on combined CRLs, while certificates that expired before 1/1/2010 will be removed.

- The **Period in months** option specifies a number of months. Revoked certificates that have been expired for longer than the specified number of months are removed from combined CRLs.

For example, if the number of months is 24, certificates that have been expired for longer than 24 months will be removed from combined CRLs.

For more information about revoking certificates, see [“Revoking user certificates” on page 174](#).

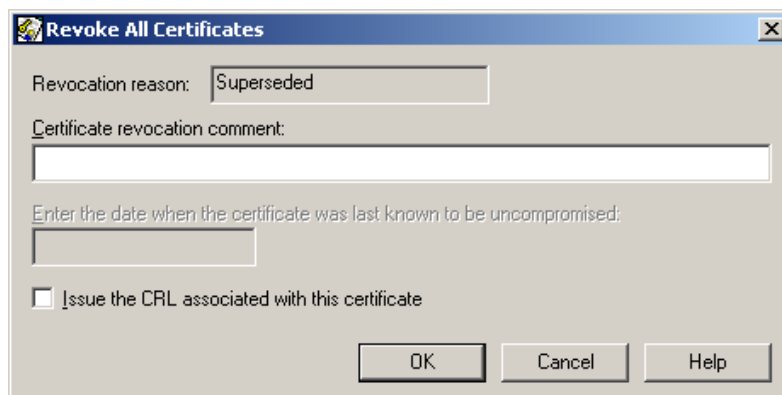
**13** Click **Apply** and authorize the operation by following the steps in [“To authorize a sensitive operation that requires one authorization” on page 53](#).

You have now set the Administration Policy.

## Interaction between CRL settings in the Security Policy


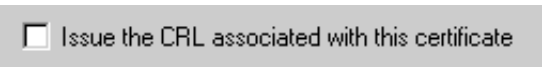


The CRL policy option **Always issue a new CRL after certificate revocation** (for more information about this option, see [Step 9 on page 86](#)) interacts with the Entrust PKI administrator permission **Force CRLs**. The result of this interaction appears in the **Revoke All Certificates** dialog box, in the **Issue the CRL associated with this certificate** check box.





The various results of this interaction, which appear in the Revoke All Certificates dialog box, are summarized in Table 3.

**Table 3:** Issue CRL check box in Revoke All Certificates dialog box

If permission	and if option	the result is
Force CRLs = Yes	Always issue a new CRL after certificate revocation = Yes	The CRL is issued with every certificate that is revoked. The check box in the <b>Revoke Certificate</b> dialog box is selected and is grayed out. 
Force CRLs = Yes	Always issue a new CRL after certificate revocation = No	The Entrust PKI administrator can choose whether to issue the CRL associated with the certificate in the <b>Revoke Certificate</b> dialog box. 
Force CRLs = No	Always issue a new CRL after certificate revocation = Yes	The CRL is issued with every certificate that is revoked. The check box in the <b>Revoke Certificate</b> dialog box is selected and is grayed out. 
Force CRLs = No	Always issue a new CRL after certificate revocation = No	The CRL is issued with every certificate that is revoked. The check box in the <b>Revoke Certificate</b> dialog box is not selected and is grayed out. 

## Configuring the encryption and verification OIDs

Encryption and verification policies are represented using object identifiers (OIDs). See the *Security Manager Deployment Guide* for conceptual information about OIDs.

Regardless of the number of policy OIDs available, you can select a maximum of 10 policy OIDs on the **Encryption OIDs** property page, and a maximum of 10 policy OIDs on the **Verification OIDs** property page. The steps for adding, deleting, and assigning OIDs are the same for both the **Encryption OIDs** property page and the **Verification OIDs** property page.

Topics in this section:

- [“Adding OIDs” on page 90](#)
- [“Deleting OIDs” on page 92](#)
- [“Adding OIDs to the default policy list” on page 93](#)
- [“Removing OIDs from the default policy list” on page 94](#)

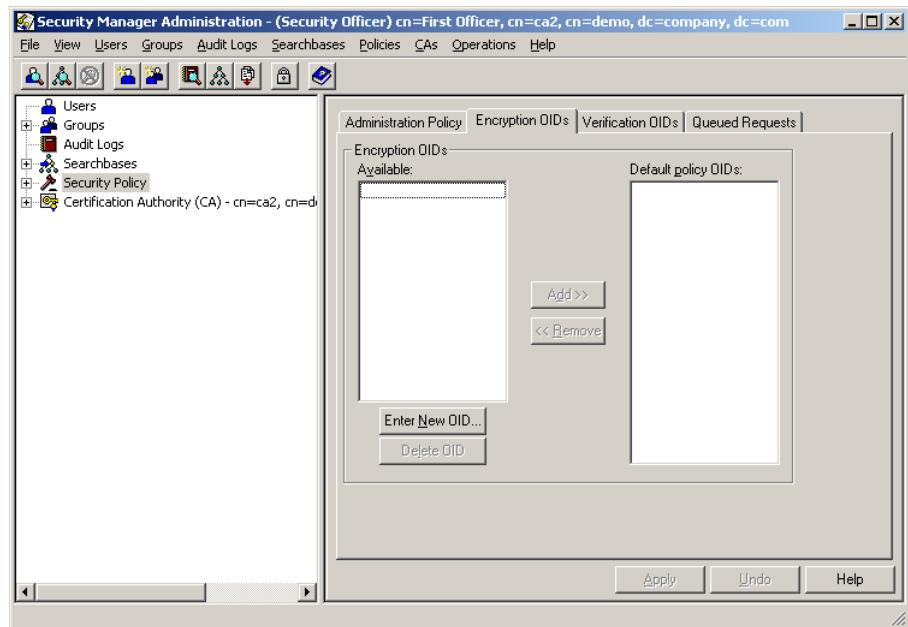
### Adding OIDs

Add OIDs according to the security needs of your organization. There is no limit to the number of OIDs you can add.

When you create a new OID, it appears in the **Available** list of both the **Encryption OIDs** property page and the **Verification OIDs** property page.

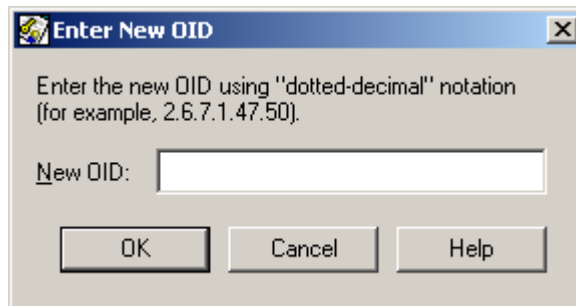
## To add an OID

- 1 Log in to Security Manager Administration as a Security Officer (see [“Logging in to Security Manager Administration”](#) on page 46).



- 2 In the tree view, click **Security Policy**.
- 3 Click the **Encryption OIDs** or the **Verification OIDs** tab to access the desired property page.
- 4 Click **Enter New OID**.

The **Enter New OID** dialog box appears.



- 5 In the **New OID** field, enter the new OID.
- 6 Click **OK**.

The new OID appears in the **Available** list.

- 7 Click **Apply** and authorize the operation by following the steps in [“To authorize a sensitive operation that requires one authorization” on page 53](#).

You have now created a new OID that appears in the **Available** list of both the **Encryption OIDs** and **Verification OIDs** property pages.

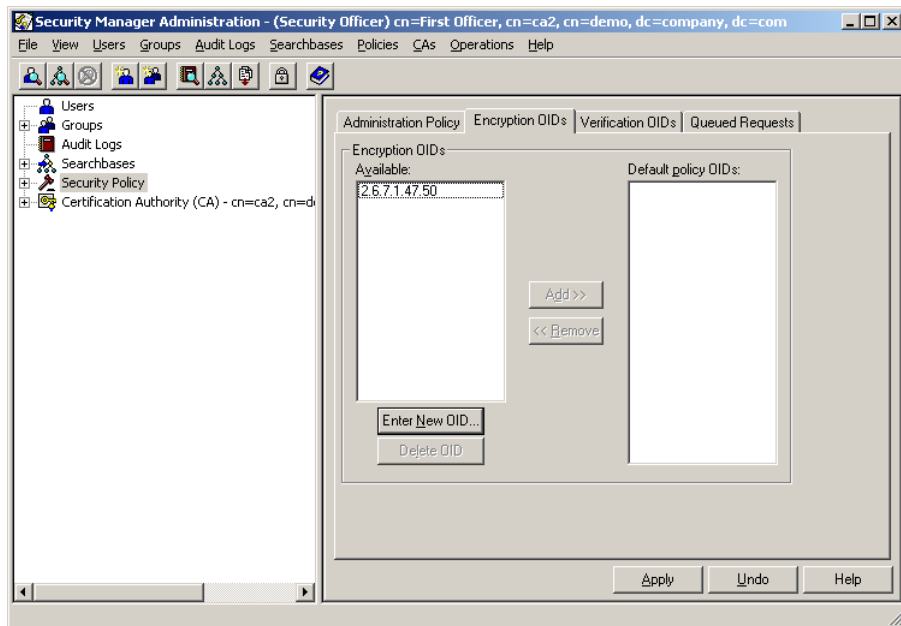
## Deleting OIDs

Delete OIDs according to the security needs of your organization.

You cannot delete an OID unless it appears in the **Available** list of both the **Encryption OIDs** and **Verification OIDs** property pages. If the OID is in the **Default policy OIDs** lists, first move it to the **Available** lists.

### To delete an OID

- 1 Log in to Security Manager Administration as a Security Officer (see [“Logging in to Security Manager Administration” on page 46](#)).



- 2 In the tree view, click **Security Policy**.
- 3 Click the **Encryption OIDs** or the **Verification OIDs** tab to access the desired property page.
- 4 In the **Available** list, click the OID that you want to delete.
- 5 Click **Delete OID**.

The OID disappears from the list.

- 6 Click **Apply** and authorize the operation by following the steps in [“To authorize a sensitive operation that requires one authorization” on page 53](#).

You have now deleted an OID from the **Available** list of both the **Encryption OIDs** and **Verification OIDs** property pages.

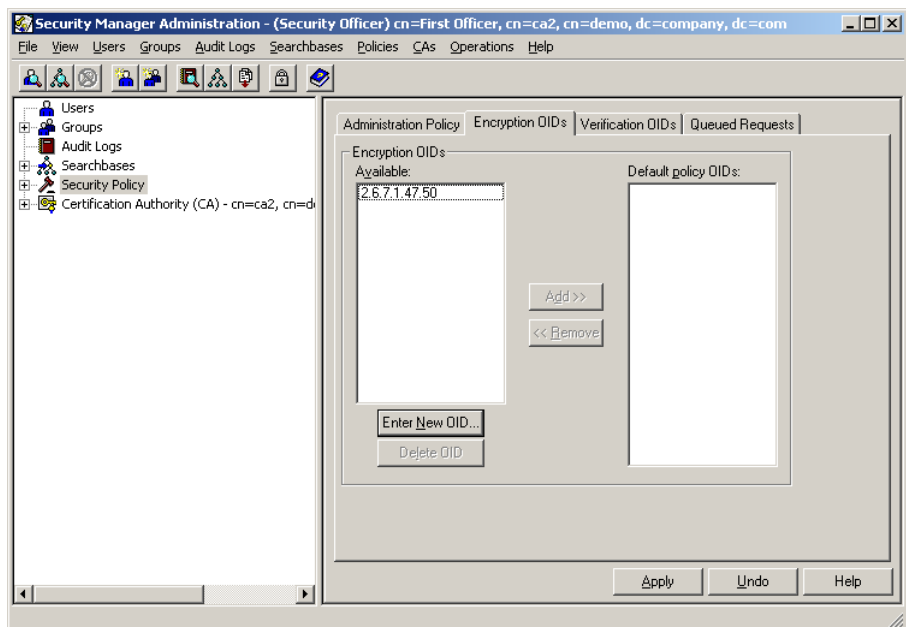
## Adding OIDs to the default policy list

When you add an OID to the **Default policy OIDs** list, it appears in the corresponding **Assigned** list in the user's **Encryption OIDs** or **Verification OIDs** property page (see [“Configuring user encryption and verification OIDs” on page 237](#)).

When you create a user's certificate using the default policy OIDs, the certificate uses the OIDs in the user's **Assigned** lists (in the user's **Encryption OIDs** and **Verification OIDs** property pages).

### To add an OID to the default policy list

- 1 Log in to Security Manager Administration as a Security Officer (see [“Logging in to Security Manager Administration” on page 46](#)).



- 2 In the tree view, click **Security Policy**.
- 3 Click the **Encryption OIDs** or the **Verification OIDs** tab to access the desired property page.
- 4 Click the OID in the **Available** list that you want to add to the **Default policy OIDs** list.

**5** Click **Add**.

The OID is added to the **Default policy OIDs** list.

**6** Click **Apply** and authorize the operation by following the steps in [“To authorize a sensitive operation that requires one authorization” on page 53](#).

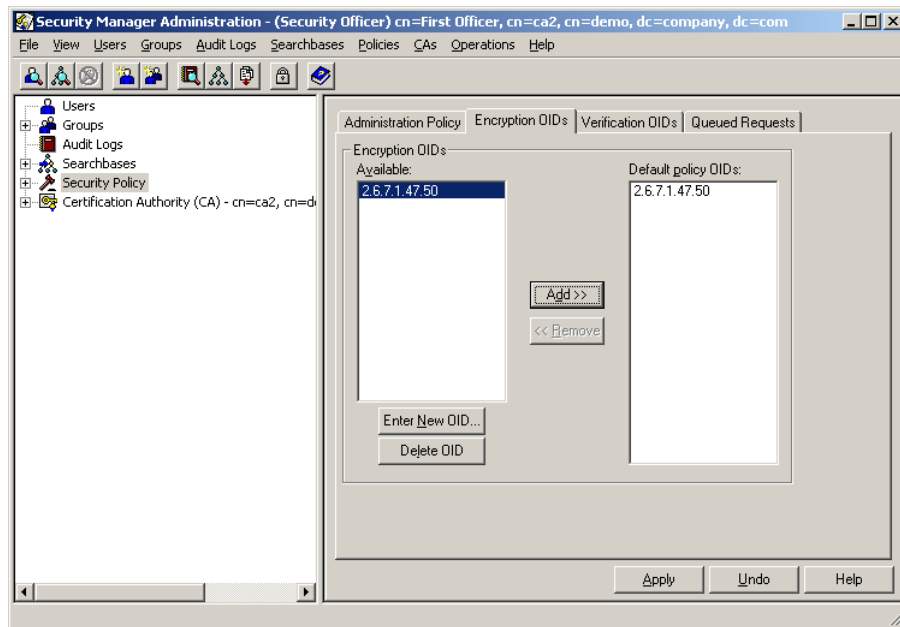
You have now added an OID to the **Default policy OIDs** list of either the **Encryption OIDs** or **Verification OIDs** property page.

## Removing OIDs from the default policy list

When you remove an OID from the **Default policy OIDs** list, it reappears in the **Available** list in the user's **Encryption OIDs** or **Verification OIDs** property page.

### To remove an OID from the default policy list

**1** Log in to Security Manager Administration as a Security Officer (see [“Logging in to Security Manager Administration” on page 46](#)).



**2** In the tree view, click **Security Policy**.

**3** Click the **Encryption OIDs** or the **Verification OIDs** tab to access the desired property page.

**4** Click the OID in the **Default policy OIDs** list that you want to move back to the **Available** list.

**5** Click **Remove**.

The OID is moved to the **Available** list.

- 6 Click **Apply** and authorize the operation by following the steps in [“To authorize a sensitive operation that requires one authorization” on page 53](#).

You have now removed an OID from the **Default policy OIDs** list of either the **Encryption OIDs** or **Verification OIDs** property page.

## Configuring queued requests

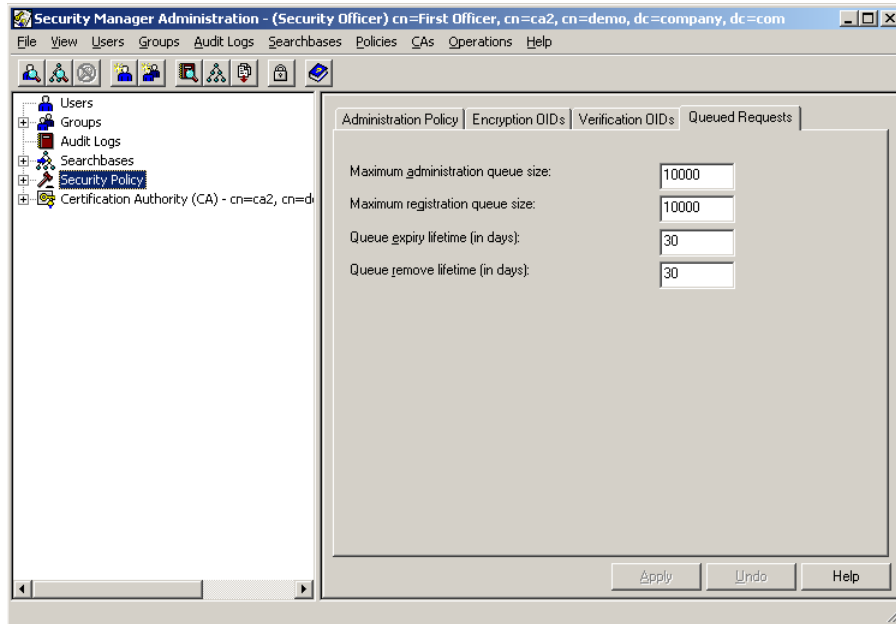
The **Queued Requests** property page allows you to configure queued requests for Security Manager client applications that support queuing (such as Administration Services). Queuing allows Entrust PKI administrators to place requests on a queue so that other administrators can authorize the operation. Security Manager Administration does not support queuing.

Besides the settings on the **Queued Requests** page, keep in mind the other settings related to queued requests:

- The `entmgr.ini` file entries `entQueueMaintPeriod`, `entQueueMaintNotBefore` and `entQueueMaintNotAfter` allow you to control when and how often automatic queue maintenance is performed. See the *Security Manager Operations Guide* for more information.
- The `entmgr.ini` file entry `entQueueSizeThreshold` allows you to control when Security Manager issues an audit to indicate that the requests queue is becoming full. See the *Security Manager Operations Guide* for more information.
- The PKI Status audit, -7642, includes the current size and maximum size of both the normal and forced queues. See the *Security Manager Operations Guide* for information about this audit log.

### To configure queued requests

- 1 Log in to Security Manager Administration as a Security Officer (see [“Logging in to Security Manager Administration” on page 46](#)).



- 2** In the tree view, click **Security Policy**.
- 3** Click the **Queued Requests** tab.
- 4** In the **Maximum administration queue size** field, enter a number from 0 to 10000000, or accept the default (10000).

The maximum administration queue size defines the maximum number of administrative requests allowed in the queue. An administrative operation is an administrator-initiated operation. If an attempt to queue an operation is made and the current queue size exceeds the maximum, a message appears indicating that the queue is full.

- 5** In the **Maximum registration queue size** field, enter a number from 0 to 10000000, or accept the default (10000).

The maximum registration queue size defines the maximum number of registration requests that can be queued. A registration operation is an end user-initiated operation. If an attempt to queue an operation is made and the current queue size exceeds the maximum, a message appears indicating that the queue is full.

- 6** In the **Queue expiry lifetime (in days)** field, enter a number from 0 to 3650, or accept the default (30).

A queue expiry lifetime setting is defined in the global security policy. Any requests that remain in the queued state longer than the lifetime have their state



set to expired. They then remain on the queue for the number of days specified in the **Queue remove lifetime (in days)** field.

- 7** In the **Queue remove lifetime (in days)** field, enter a number from 0 to 3650, or accept the default (30).

Requests remain on the queue for this amount of time after they complete, or have failed, expired, or been cancelled.

- 8** Click **Apply**.

You have now configured queued requests.

## Configuring options for the various certificate categories

Certificates are put into the following categories:

- Enterprise
- Web
- Cross-certificates
- CA certificates

Enterprise and Web **Certificate Categories** appear under **Security Policy** in the tree view. (If you did not purchase a Web license, you do not use Web certificates and they do not appear in the tree view.) Although cross-certificates make up a certificate category according to the certificate specifications, cross-certificates are only given a lifetime and do not have any other options you can select.

For each of the certificate categories that appear in your tree view (for example, Enterprise), you must set the following options:

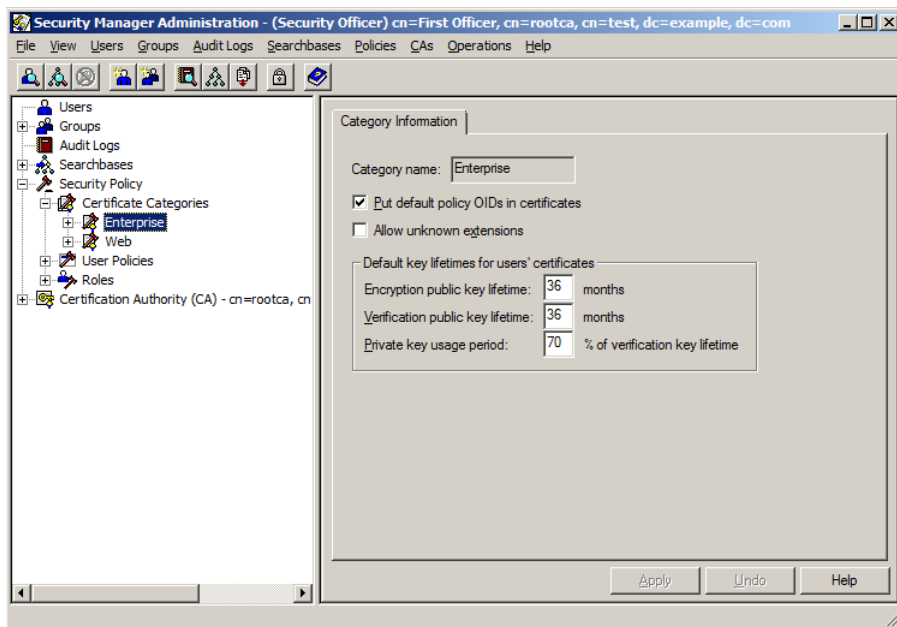
- whether to include default policy OIDs in certificates
- whether the certificate category allows unknown extensions
- the default key lifetimes for users' certificates

This topic contains the following procedures:

- [“To set options for the Enterprise certificate category” on page 97](#)
- [“To set options for the Web certificate category” on page 100](#)

### To set options for the Enterprise certificate category

- 1** Log in to Security Manager Administration as a Security Officer (see [“Logging in to Security Manager Administration” on page 46](#)).



- 2 In the tree view, expand **Security Policy** and then expand **Certificate Categories**.  
A list of certificate categories appears in the tree view. (Only those categories for which you have purchased a license appear, such as **Enterprise**.)
- 3 Select the **Enterprise** certificate category.  
The **Category Information** property page appears.
- 4 If desired, select **Put default policy OIDs in certificates**.  
This option interacts with the default policy OIDs defined in **Security Policy** in the tree view, policy OIDs defined in the `master.certspec` file, and policy OIDs assigned to the user. Table 4 describes the interaction.

**Table 4:** Effect of OIDs settings on certificates

If Put default policy OIDs in certificates option is	and the User's Default Policy OIDs <sup>1</sup> option is	the result is
Off	On	Policy OIDs defined in the <code>master.certspec</code> file are put into users' certificates. However, OIDs defined in <b>Security Policy</b> in the tree view are not put into user's certificates.

**Table 4:** Effect of OIDs settings on certificates (continued)

If Put default policy OIDs in certificates option is	and the User's Default Policy OIDs <sup>1</sup> option is	the result is
On	On	Policy OIDs defined in the <code>master.certspec</code> file are merged with the OIDs defined in <b>Security Policy</b> in the tree view, and are put into user's certificates.
On	Off	The policy OIDs defined for the user are used. <sup>2</sup>

1. OIDs defined in **Security Policy** in the tree view means OIDs that appear in the **Default policy OIDs** lists in the **Encryption OIDs** and **Verification OIDs** property pages.

2. These OIDs are defined in the **Assigned** lists in the user's **Encryption OIDs** and **Verification OIDs** property pages.

- 5** To allow Security Manager to sign certificates that contain extensions it does not recognize, select **Allow unknown extensions**.

Security Manager recognizes most valid extensions. Only select this option if you need to support custom extensions. Custom extensions may come from custom client applications.

- 6** In the **Encryption public key lifetime** and **Verification public key lifetime** fields, enter values between 2 and 420 (35 years), or accept the default values.

The default lifetimes apply to all user certificates (which include certificates for Security Officers, Administrators, Auditors, Directory Administrators, and End Users). You can customize the certificate lifetimes for individual users, or you can use the default settings (which you set in the **Category Information** property page). Changing certificate lifetimes does not affect any existing certificates; it only affects the lifetimes of certificates created after the modification.

Regardless of the certificate and key lifetimes you choose, Entrust desktop applications attempt to update their key pairs according to the description in the *Security Manager Deployment Guide*. Once a certificate expires, you can no longer use it.

---

**Note:** It is possible to increase the certificate lifetime beyond the maximum of 420 months up to the lifetime of the current CA certificate through the certificate definition policy by customizing the `master.certspec` file. See [“Customizing certificates” on page 525](#). If the user certificate lifetime exceeds the latest CA certificate lifetime, it is truncated to the CA certificate lifetime.

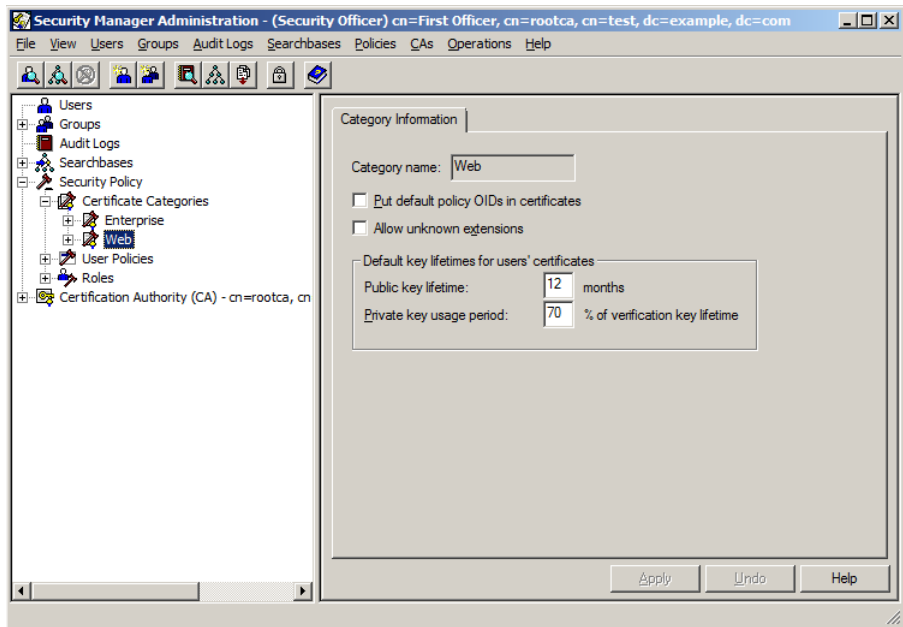
---

- 7** In the **Private key usage period** field, enter a percentage from 1 to 100, or accept the default (70%).

You have now set the options for certificate categories.

## To set options for the Web certificate category

- 1 Log in to Security Manager Administration as a Security Officer (see [“Logging in to Security Manager Administration”](#) on page 46).



- 2 In the tree view, expand **Security Policy** and then expand **Certificate Categories**.  
A list of certificate categories appears in the tree view. (Only those categories for which you have purchased a license appear, such as **Web**.)
- 3 Select the **Web** certificate category.  
The **Category Information** property page appears.
- 4 If desired, select **Put default policy OIDs in certificates**.  
This option interacts with the default policy OIDs defined in **Security Policy** in the tree view, policy OIDs defined in the `master.certspec` file, and policy OIDs assigned to the user. Table 5 describes the interaction.

**Table 5:** Effect of OIDs settings on certificates

If Put default policy OIDs in certificates option is	and the User's Default Policy OIDs <sup>1</sup> option is	the result is
Off	On	Policy OIDs defined in the <code>master.certspec</code> file are put into users' certificates. However, OIDs defined in <b>Security Policy</b> in the tree view are not put into user's certificates.

**Table 5:** Effect of OIDs settings on certificates (continued)

If Put default policy OIDs in certificates option is	and the User's Default Policy OIDs <sup>1</sup> option is	the result is
On	On	Policy OIDs defined in the <code>master.certspec</code> file are merged with the OIDs defined in <b>Security Policy</b> in the tree view, and are put into user's certificates.
On	Off	The policy OIDs defined for the user are used. <sup>2</sup>

1. OIDs defined in **Security Policy** in the tree view means OIDs that appear in the **Default policy OIDs** lists in the **Encryption OIDs** and **Verification OIDs** property pages.

2. These OIDs are defined in the **Assigned** lists in the user's **Encryption OIDs** and **Verification OIDs** property pages.

- 5** To allow Security Manager to sign certificates that contain extensions it does not recognize, select **Allow unknown extensions**.

Security Manager recognizes most valid extensions. Only select this option if you need to support custom extensions. Custom extensions may come from custom client applications.

- 6** In the **Public key lifetime** field, enter a value between 2 and 420 (35 years), or accept the default value.

The default lifetimes apply to all user certificates (which include certificates for Security Officers, Administrators, Auditors, Directory Administrators, and End Users). You can customize the certificate lifetimes for individual users, or you can use the default settings (which you set in the **Category Information** property page). Changing certificate lifetimes does not affect any existing certificates; it only affects the lifetimes of certificates created after the modification.

Regardless of the certificate and key lifetimes you choose, Entrust desktop applications attempt to update their key pairs according to the description in the *Security Manager Deployment Guide*. Once a certificate expires, you can no longer use it.

---

**Note:** It is possible to increase the certificate lifetime beyond the maximum of 420 months up to the lifetime of the current CA certificate through the certificate definition policy by customizing the `master.certspec` file. See [“Customizing certificates” on page 525](#). If the user certificate lifetime exceeds the latest CA certificate lifetime, it is truncated to the CA certificate lifetime.

---

- 7** In the **Private key usage period** field, enter a percentage from 1 to 100, or accept the default (70%).

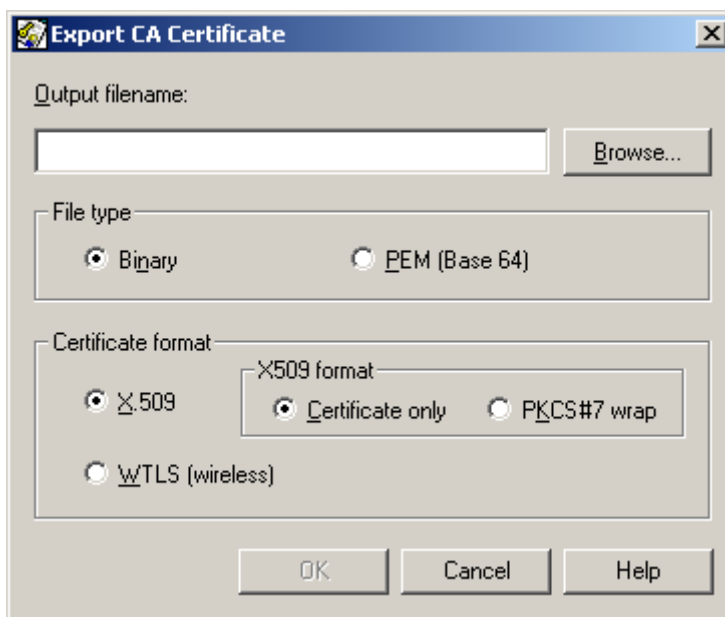
You have now set the options for certificate categories.

# Exporting the current CA certificate

Some applications may require a CA certificate from other applications as a prerequisite for working with them. If an application you want Security Manager to work with has this requirement, you can meet it by exporting a CA certificate to a file and submitting the file to the other application.

## To export the CA certificate

- 1 Log in to Security Manager Administration as a Security Officer (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Select **CAs > Export CA Certificate**.
- 3 Select **Export CA Certificate** in the pop-up menu.
- 4 In the **Export CA Certificate** dialog box, type a filename in the **Output filename** field, and click **Browse** to specify a location for the file.

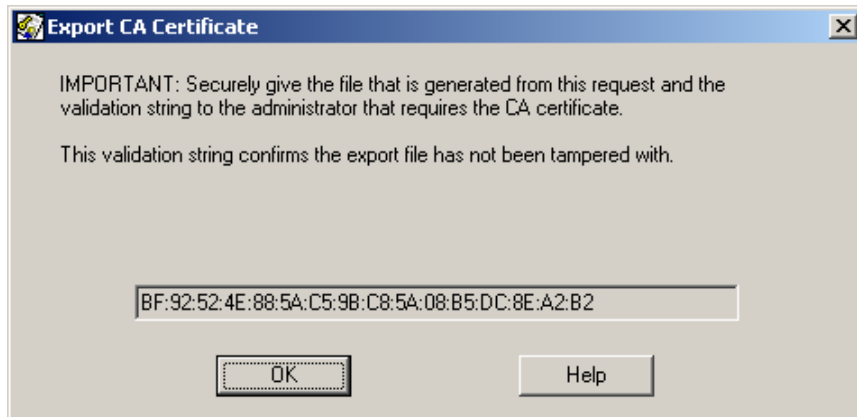


- 5 Select either **Binary** or **PEM (Base 64)** for **File type**.  
If you select **Binary**, an extension of `.der` appends to the filename. If you select **PEM (Base 64)**, an extension of `.pem` appends to the filename. The choice of file formats is provided for interoperability with PKIs from other vendors. Security Manager can read certificates exported in either file format.
- 6 Select **WTLS (wireless)** for **Certificate format** if you are exporting the CA certificate for use with WAP servers; otherwise choose **X.509**.

If you are exporting the CA as an X.509 certificate, you must choose to export it either as raw certificate data (**Certificate only**), or you can export it using the PKCS #7 standard (**PKCS#7 wrap**). In a scenario where there is a hierarchy of CAs, the raw certificate data only contains information about the certificate you are exporting, whereas the PKCS #7 format includes the certificate chain, including any superior CAs, if the certificate you are exporting is from a subordinate CA.

**7** Click **OK**.

When the CA certificate completes writing to file, a dialog box appears displaying the validation string for the exported CA certificate.



Later when the exported CA certificate is imported elsewhere (for example, on a WAP server, or by a Security Officer at another CA), this validation string is used to confirm that the exported CA certificate has not been altered.

If your CA is a root CA, the validation string is based on an MD5 hash of the CA certificate, and the validation string is the same as the CA's Web fingerprint. If your CA is a subordinate CA, the validation string is based on an MD5 hash of the root CA, and is the same value as the root CA's Web fingerprint, not your CA's Web fingerprint. For information about viewing the Web fingerprint, see ["Viewing CA information" on page 104](#).

**8** Record the validation string on paper and click **OK**.

You have now exported the CA certificate.

# Viewing CA information

In Security Manager Administration, you can view general information about a CA, or detailed information about the CA certificate.

The general information includes the CA's DN, the DN of the CA that issued the CA certificate (either the CA's DN or a superior CA's DN if there is one), the Web fingerprint, the key pair algorithm and size, and the hardware identifier if the CA key is stored on a hardware device. The Web fingerprint is a string of alphanumeric characters based on an MD5 hash of the latest CA root certificate.

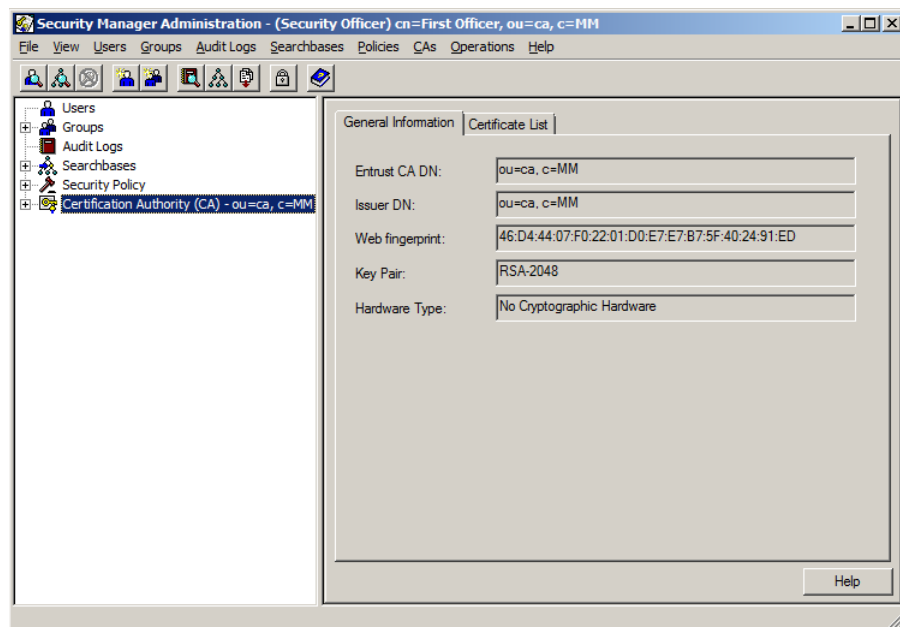
Security Manager Administration also allows you to view detailed information about a CA certificate, the contents of the CA certificate decoded in ASN.1 format, and the certificate chain. When you view the contents of a CA certificate, you can see all the details of the actual certificate, some of which are not displayed anywhere else. For example, if you want to see all the extensions in a CA certificate, you can do so by viewing the certificate contents.

This section contains the following procedures:

- [“To view general information about the CA” on page 104](#)
- [“To view the contents of a CA certificate” on page 105](#)

## To view general information about the CA

- 1 Log in to Security Manager Administration as a Security Officer (see [“Logging in to Security Manager Administration” on page 46](#)).



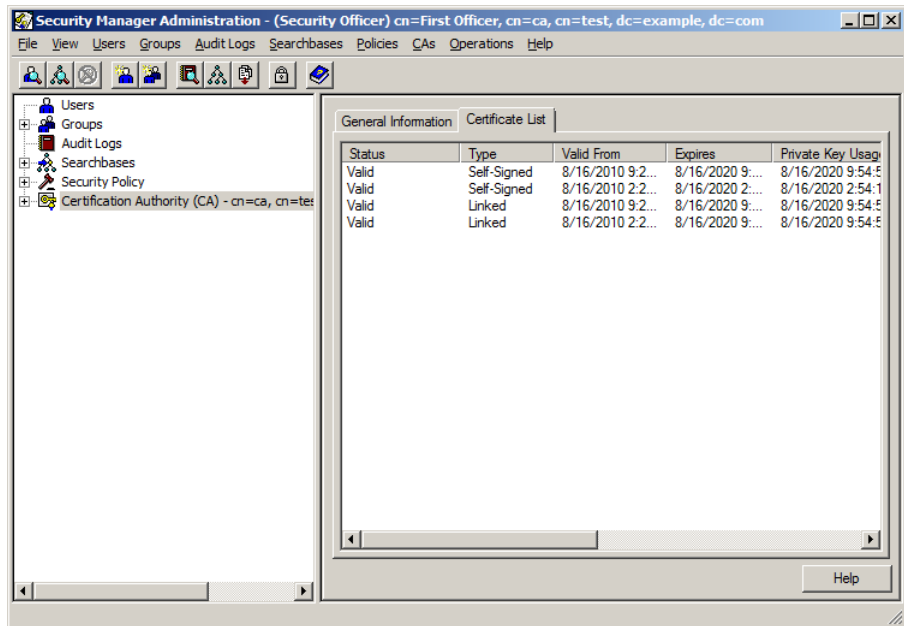


- 2 In Security Manager Administration, click the CA certificate icon in the tree view (for example, **Certification Authority (CA) - dc=Company One,dc=com**).
- 3 Click the **General Information** tab.

You have now viewed general information about the CA.

### To view the contents of a CA certificate

- 1 Log in to Security Manager Administration as a Security Officer (see [“Logging in to Security Manager Administration”](#) on page 46).



- 2 In Security Manager Administration, click the CA certificate icon in the tree view (for example, **Certification Authority (CA) - dc=Company One,dc=com**).
- 3 Click the **Certificate List** tab.
- 4 Double-click the CA certificate for which you want to view the contents, or right-click the CA certificate and select **Certificate Contents** from the pop-up menu.

The **Certificate** dialog box appears. The **Certificate** dialog box has three tabs:

- **Details**

The **Details** property page contains certificate information, including the certificate definition name from the certificate specification, the serial number of the certificate, the issuing CA, the encryption algorithm, the extensions in the certificate, and status information.

- **Contents**

The **Contents** property page contains an ASN.1 decoding of the certificate.

- **Certificate Chain**

The **Certificate Chain** property page contains the chain of issuing CAs, from the CA that signed the certificate to the root CA.

**5** When you have finished viewing the contents, click **Close**.

You have now viewed the contents of a CA certificate.

## Configuring users' key pairs

Security Manager provides a flexible system of key-pair assignment that allows you to select the best key-pair model for a user or group of users.

This chapter explains when to assign each type of model, and the process involved in configuring the user's key pairs. The chapter includes the following sections:

- ["Overview of key pairs" on page 108](#)
- ["Supported key-pair models" on page 113](#)
- ["Configuring key-pair users" on page 118](#)

# Overview of key pairs

Security Manager allows you to create five different types of key pairs:

- dual-usage
- encryption
- signing
- nonrepudiation
- EFS encryption

The following sections describe these key pairs:

- [“The dual-usage key pair” on page 108](#)
- [“The encryption key pair” on page 109](#)
- [“The signing key pair” on page 110](#)
- [“The nonrepudiation key pair” on page 110](#)
- [“The EFS encryption key pair” on page 111](#)

## The dual-usage key pair

Characteristics of dual-usage key pairs are:

- The keys are created by a Security Manager client application on the end user's local computer.
- The certificate is created by Security Manager and available in the directory and the user's Entrust digital ID.
- The public key is publicly available within the directory and stored in the user's Entrust digital ID.
- The dual-usage private key is placed in the user's Entrust digital ID.

This allows anyone to encrypt data for the user using the dual-usage public key stored in the directory, whereas the user is the only person who can decrypt that data using the dual-usage private key stored in the digital ID.

- Both of these keys and the certificate are backed up in order to provide a key history.

For example, if an administrator changes the encryption algorithm and issues a new dual-usage public key for the user, it is the key history, backed up and maintained in the Security Manager database, that allows the user to continue to access data that was encrypted using one of the previous dual-usage public keys.

[Table 6 on page 109](#) shows what keys and certificates are created when a user first creates an Entrust profile, and where those keys and certificates are stored and backed up.

**Table 6:** Creation and location of single key pair

Key	Created by	Is put in	Backup details
dual-usage private key	client application	user's Entrust profile	backed up securely in the Security Manager database
dual-usage public key	client application	dual-usage public key certificate	backed up whenever dual-usage public key certificate is backed up, in the Security Manager database
dual-usage public key certificate	Security Manager	directory and the user's Entrust profile	backed up in the Security Manager database

## The encryption key pair

In this model, Security Manager typically creates the encryption key pair. The encryption public key is usually stored in the directory, and the decryption private key is part of the user's digital ID. Security Manager backs up both keys in the database to maintain a key history. This allows users to access data that was encrypted using one of the previous encryption key pairs.

Table 7 shows what keys and certificates related to the encryption key pair are created when a user first creates an Entrust profile, and where those keys and certificates are stored and backed up.

**Table 7:** Creation and location of encryption keys

Key	Created by	Is put in	Backup details
decryption private key	Security Manager	user's Entrust profile	backed up securely in the Security Manager database
encryption public key	Security Manager	encryption public key certificate	backed up whenever the encryption public key certificate is backed up, in the Security Manager database
encryption public key certificate	Security Manager	directory and the user's Entrust profile	backed up in the Security Manager database

# The signing key pair

A client application (such as Security Provider for Windows) creates the signing key pair on the user's computer. The signing private key never leaves the end user's computer and is not backed up anywhere. The public verification key is included with all signed data and backed up in the Security Manager database.

The signing private key is not sent to Security Manager and, therefore, not backed up in the Security Manager database.

When Security Manager receives the verification public key, Security Manager creates a verification public key certificate for the verification public key. A copy of the verification public key certificate is stored in the Security Manager database and a copy is returned to the Entrust desktop application.

Unlike the encryption public key certificate, a copy of the verification public key certificate is not stored in the user's directory entry. When a user signs a file using an Entrust desktop application, it includes the verification public key certificate with the signed file. Therefore, retrieval of the verification public key certificate from the directory is never required.

Table 8 shows what keys and certificates related to the signing key pair are created when a user first creates an Entrust profile, and where those keys and certificates are stored and backed up.

**Table 8:** Creation and location of signing keys

Key	Created by	Is put in	Backup details
signing private key	client application	user's Entrust profile	N/A
verification public key	client application	verification public key certificate	backed up in the Security Manager database
verification public key certificate	Security Manager	user's Entrust profile and Security Manager database	N/A

# The nonrepudiation key pair

Though the signing key provides nonrepudiation, it is also used for authentication purposes to the CA when requesting a key update. Therefore, if you have a high-assurance user, you may want to use a nonrepudiation key pair that specifically provides nonrepudiation for highly important information.

When you are using an Entrust security store, a user must re-authenticate before using the nonrepudiation key pair to sign the data. This adds an extra layer of nonrepudiation not provided by a normal signing key pair.

An Entrust client application creates the nonrepudiation keys, which are stored exclusively on the user's computer. To ensure nonrepudiation, the private key in the nonrepudiation key pair never leaves the user's computer and is never backed up.

When Security Manager receives the nonrepudiation public key, Security Manager creates a nonrepudiation public key certificate. A copy of the nonrepudiation public key certificate is stored in the Security Manager database and a copy is returned to the Entrust desktop application.

**Table 9:** Creation and location of nonrepudiation keys

Key	Created by	Is put in	Backup details
nonrepudiation private key	client application	user's Entrust profile	N/A
nonrepudiation public key	client application	nonrepudiation public key certificate	backed up in the Security Manager database
nonrepudiation public key certificate	Security Manager	user's Entrust profile and Security Manager database	N/A

## The EFS encryption key pair

The EFS encryption key pair is used when users need to encrypt and decrypt data while interoperating with the Microsoft File Encryption System (EFS).

An Entrust client application creates the EFS encryption keys. The EFS key pair is backed up in the database and the public encryption key is stored in the directory.

Table 10 shows what keys and certificates related to the encryption key pair are created when a user first creates an Entrust digital ID, and where those keys and certificates are stored and backed up.

**Table 10:** Creation and location of EFS encryption keys

Key	Created by	Is put in	Backup details
EFS decryption private key	client application	user's Entrust security store	backed up securely in the Security Manager database
EFS encryption public key	client application	encryption public key certificate	backed up whenever the encryption public key certificate is backed up, in the Security Manager database

**Table 10:** Creation and location of EFS encryption keys (continued)

Key	Created by	Is put in	Backup details
EFS encryption public key certificate	Security Manager	directory and the user's Entrust profile	backed up in the Security Manager database



# Supported key-pair models

Security Manager has a flexible system that allows you to assign users a variety of key-pair models, ranging from one key pair to four key pairs. The key-pair model you choose for a particular user or group of users depends on these factors:

- the users' authentication, confidentiality, and nonrepudiation requirements (key usage)
- their use of third-party data security applications, such as Secure Multi-purpose Internet Mail Extension (S/MIME) or Microsoft Encryption File System (EFS)
- the specific client application they will be using (for more information about an application's supported key-pair models, consult the Entrust application's documentation)

For more information about available key pairs, see [“Overview of key pairs” on page 108](#).

Consult [Table 11 on page 113](#) and the sections below to decide which model fits the users' needs.

**Table 11:** Determining the key-pair model

Key usage	Third-party application	Entrust application	Model
Encryption and digital signature	S/MIME	V1	V1 1-key-pair
Encryption and digital signature or nonrepudiation	S/MIME	V2	V2 1-key-pair
Encryption and digital signature		V1	V1 2-key-pair
Encryption and digital signature		V2	V2 2-key-pair
EFS encryption and CMP signing	EFS, no email encryption or signing needs <sup>1</sup>	V2	V2 Standalone EFS 2-key-pair
Encryption, digital signature, and EFS encryption	EFS	V2	V2 EFS 3-key-pair
Encryption, digital signature, and nonrepudiation		V2	V2 Nonrepudiation 3-key-pair

**Table 11:** Determining the key-pair model (continued)

Key usage	Third-party application	Entrust application	Model
Encryption, digital signature, EFS encryption, and nonrepudiation	EFS	V2	V2 4-key-pair

1. The V2 Standalone EFS 2-key-pair model includes two key pairs, one for EFS encryption, and one for Certificate Management Protocol (CMP) signing. The CMP signing key pair is used only to sign messages to the CA, since there is no need to sign email messages. In fact, the CMP signing key contains an OID that prevents third-party CAPI-enabled applications from using it as a signing key.

The V2 key-pair models are not compatible with V1 Entrust applications. For more information on V1 and V2, refer to the *Security Manager Deployment Guide*.

If a user wants to change from using a V2 application (such as Entrust Entelligence Security Provider) to a V1 application (such as Entrust Entelligence Desktop Solutions), the first step is to set the user for key recovery. After that, a Master User can change their status in Security Manager Control Command Shell (see the *Security Manager Operations Guide*). When their status is changed in Security Manager, you can provide the reference number and authorization code so that the user can recover their digital ID. You should be aware that the conversion of the digital ID data to V1 means that the user cannot access data in existing V2-key-pair Entrust security stores.

Topics in this section:

- [“1-key pair option” on page 114](#)
- [“2-key pair option” on page 115](#)
- [“3-key pair option” on page 116](#)
- [“4-key pair option” on page 117](#)

## 1-key pair option

Use the 1-key pair (also referred to as dual-usage keys) option if:

- your users require either encryption or signing
- you do not mind if the same key pair is used for both encryption and signing
- your third-party S/MIME product does not support two or more key pairs

Because the 1-key pair is used for both encryption and signing, Security Manager backs up both the private and public key in its database. Therefore, you cannot use 1-key pairs for nonrepudiation, as there is more than one copy of the signing key.

---

**Attention:** It is recommended that when nonrepudiation is an essential feature of your security policy, you do not create 1-key-pair users.

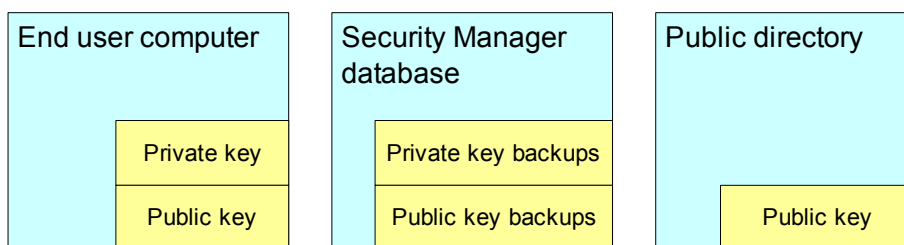
---

Do not use the 1-key pair option if:

- Both digital signature and nonrepudiation are essential, since the dual-usage private key will be backed up in order to enable key recovery.
- The user is an Entrust PKI administrator. While users of any role can be 1-key-pair users, nonrepudiation is essential for Entrust PKI administrators.

By default, Security Manager can create two types of 1-key pairs—one for V1 digital IDs, the other for V2 digital IDs. In the V1 1-key-pair model, the encryption and digital signature functions are combined. The V2 1-key-pair model combines the encryption function with either the digital signature function or the nonrepudiation function.

**Figure 1:** A typical 1-key pair model



The V1 and V2 2-key-pair models are designed for the most common encryption and digital signature requirements. They take a fairly typical approach to security requirements by providing the encryption and digital signature capabilities separated into two key pairs.

## 2-key pair option

Use the 2-key pair option if:

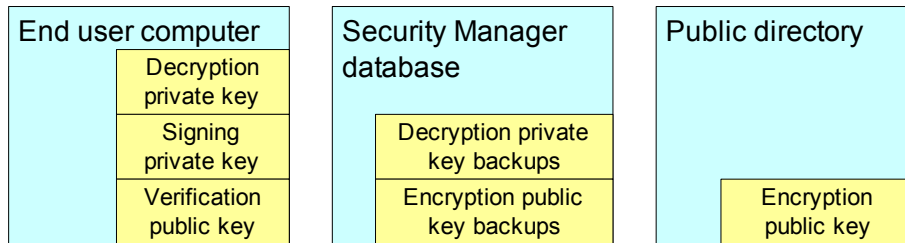
- your users require encryption and signing (verification)
- you want different key pairs to provide this functionality to ensure nonrepudiation
- you have users who need EFS encryption, but do not need email encryption or signing capabilities (use the V2 Standalone EFS 2-key-pair option)

Do not use the 2-key pair option if:

- users have third-party S/MIME applications. In this case, you should consider a 1-key-pair model.
- users have high-assurance positions, or when they are users of the Microsoft Encryption File System (EFS). In these cases, you should consider a 3-key-pair or 4-key-pair model.

By default, Security Manager can create 2-key pair digital IDs for both V1 and V2 clients.

**Figure 2:** A typical 2-key pair model



## 3-key pair option

Use a 3-key pair digital ID if:

- you want to define different keys for encryption, signing, and nonrepudiation or EFS encryption
- you have users in high-assurance positions who need to have separate digital signature and nonrepudiation keys
- you have users of the Microsoft Encryption File System (EFS) application, who also need email encryption and signing capabilities

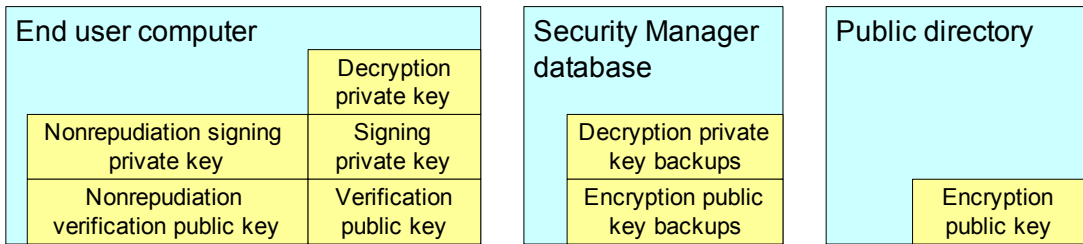
Though the signing key provides nonrepudiation, it is also used for authentication purposes to the CA when requesting a key update. Therefore, if you have a high-assurance user, you may want to add a third key pair that specifically provides nonrepudiation for highly important information.

When you are using an Entrust security store, a user must re-authenticate before using the nonrepudiation key pair to sign the data. This adds an extra layer of nonrepudiation not provided by a normal signing key pair.

Alternatively, your users may require encryption, signing, and EFS encryption capabilities. If so, you can use a 3-key pair model that would create a key pair for each option.

If you require 3-key pair digital IDs, you can only use V2-compliant clients. V1 clients are limited to 1 or 2-key pairs.

**Figure 3:** A typical 3-key pair model with a nonrepudiation key pair



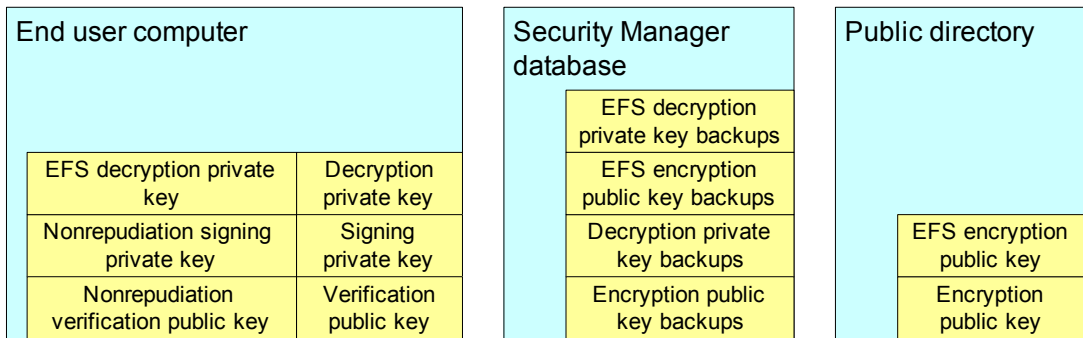
## 4-key pair option

Use a 4-key pair if you want to define different key pairs for encryption, signing, nonrepudiation, and EFS encryption.

The 4-key-pair model is designed for the high-assurance users who are also EFS users.

If you require 4-key pair digital IDs, you can only use V2-compliant clients. V1 clients are limited to 1 or 2-key pairs.

**Figure 4:** A typical 4-key pair model



# Configuring key-pair users

Once you have determined that a user or group of users need a particular key-pair model, you can begin configuring the model. (For information to help you make these decisions, see [“Supported key-pair models” on page 113.](#))

How you configure a key-pair user depends on the key-pair model you choose. Table 12 shows how to find the procedures for configuring each model.

**Table 12:** Procedures for configuring key-pair models

To configure this key-pair model...	Go to...
V1 1-key-pair	<a href="#">“Configuring V1 1-key-pair users” on page 119</a>
V1 2-key-pair	<a href="#">“Creating new users” on page 146</a>
V2 1-key-pair Any V2 2-key-pair Any V2 3-key-pair V2 4-key-pair	<a href="#">“Configuring V2 key-pair users” on page 122</a>

Topics in this section:

- [“Configuring V1 1-key-pair users” on page 119](#)
- [“Configuring V2 key-pair users” on page 122](#)

## Configuring V1 1-key-pair users

In Security Manager, you can create Entrust PKI users with only one key pair in their V1 Entrust profile, instead of two key pairs. These users are referred to as V1 1-key-pair users. The single key pair is used for encryption, decryption, verification, and signing operations.

For more information about Entrust's single-key-pair models, see [“Supported key-pair models” on page 113](#).

---

**Note:** Users with client applications compatible with Security Manager 6.0 or earlier cannot encrypt for 1-key-pair users created in Security Manager. If you do not plan to upgrade these applications, it is recommended that you do not create 1-key-pair users.

---

Topics in this section:

- [“Creating 1-key-pair users” on page 119](#)
- [“Changing 2-key-pair users into 1-key-pair users” on page 120](#)
- [“Changing 1-key-pair users into 2-key-pair users” on page 121](#)

---

**Attention:** Before you create a V2-key-pair user with one key pair, see [“Supported key-pair models” on page 113](#).

---

### Creating 1-key-pair users

Complete the following procedure to create a 1-key-pair user with a V1-compatible digital ID.

#### To create a new 1-key-pair user

- 1 Create a new user policy for 1-key-pair users.

It is recommended that a new user policy be created based on the existing End User policy. The user policy must include the following policy settings:

- Set **Number of key pairs** to 1
- Set **Algorithm for digital signature** to an RSA value (RSA-1024, RSA-2048, RSA-3072, RSA-4096, or RSA-6144).

Entrust clients prior to Security Manager 7.0 are not compatible with RSA-4096 or RSA-6144. Entrust clients prior to Security Manager 8.0 are not compatible with RSA-3072.

A Security Officer (or other Entrust PKI administrator with appropriate permissions) must perform this step. To learn how to edit a user policy, see [“Administering user policies” on page 391](#).

---

**Attention:** If you choose an algorithm other than RSA, users you create with this user policy will be unable to create their profiles. Of the available algorithms, only RSA supports both encryption and signing.

---

- 2** Create a new user role and associate the 1-key-pair user policy with it.  
It is recommended that you create the new role on the existing End User role. An Entrust PKI administrator can perform this step. To learn how to create a new role, see [“Creating roles” on page 361](#).
- 3** Create new users and assign them the single-key-pair role. See [“Creating new users” on page 146](#).

## Changing 2-key-pair users into 1-key-pair users

Complete the following procedure to change a 2-key-pair user with a V1-compatible digital ID into a 1-key-pair user with a V1-compatible digital ID.

### To change a 2-key-pair user into a 1-key-pair user

Do one of the following:

- Change the 2-key-pair user’s role to the role you have specified as a 1-key-pair user role (see [“Configuring user properties” on page 219](#)).
- Edit the user policy associated with the role shared by all users you want to change to 1-key-pair users.

You must set **Number of key pairs** to 1 and **Algorithm for digital signature** to an RSA value (RSA-1024, RSA-2048, RSA-3072, RSA-4096, RSA-6144).

Entrust clients prior to Security Manager 7.0 are not compatible with RSA-4096 or RSA-6144. Entrust clients prior to Security Manager 8.0 are not compatible with RSA-3072.

To learn how to edit a user policy, see [“Administering user policies” on page 391](#).

---

**Attention:** If you choose an algorithm other than RSA, users you create with this user policy will be unable to create their profiles. Of the available algorithms, only RSA supports both encryption and signing.

---

For this change to take effect in the user’s profile, you must wait until the user’s keys update automatically (see [“Configuring user key update options” on page 230](#)), unless you take one of the following actions:

- Update the user’s key pairs (see [“Updating key pairs” on page 240](#)).
- Recover the user’s keys (see [“Recovering user key pairs” on page 162](#)).



- Initiate a DN change for the user (see [“Changing distinguished names” on page 193](#)).

Before changing the user from a 2-key-pair user to a 1-key-pair user, you can also revoke the user's encryption certificate. This ensures that the Entrust desktop application generates a new key pair for the user and requests a new dual-usage certificate from Security Manager. See [“Revoking user certificates” on page 174](#).

## Changing 1-key-pair users into 2-key-pair users

Complete the following procedure to change a 1-key-pair user with a V1-compatible digital ID into a 2-key-pair user with a V1-compatible digital ID.

### To change a 1-key-pair user into a 2-key-pair user

Do one of the following:

- Change the 1-key-pair user's role to the role you have specified as a 2-key-pair user role (see [“Configuring user properties” on page 219](#)).  
An Entrust PKI administrator can perform this task. With this method, you can change only one user at a time.
- Edit the user policy associated with the role shared by all users you want to change to 2-key-pair users by setting **Number of key pairs** to 2. To learn how to edit a user policy, see [“Administering user policies” on page 391](#). A Security Officer (or other Entrust PKI administrator with appropriate permissions) must perform this task.

---

**Attention:** The second method affects every user of every role that uses the modified user policy, as well as every role you assign the modified user policy to in the future.

---

For this change to take effect in the user's profile, you must wait until the user's keys update automatically (see [“Configuring user key update options” on page 230](#)), unless you take one of the following actions:

- Update the user's key pairs (see [“Updating key pairs” on page 240](#)).
- Recover the user's keys (see [“Recovering user key pairs” on page 162](#)).
- Initiate a DN change for the user (see [“Changing distinguished names” on page 193](#)).

## Configuring V2 key-pair users

When you select one of the V2 key-pair models for your users, there are two different paths you can follow to configure the models:

- In the default path, you select one of the pre-defined certificate definitions and its associated policy.
- In the customized path, you can alter a certificate definition or policy, or create a new one to suit the specific needs of your users.

Topics in this section:

- [“Which path to choose?” on page 122](#)
- [“Default configuration path” on page 126](#)
- [“Customized configuration path” on page 127](#)

### Which path to choose?

The first thing you need to do is decide whether your users fit exactly into any of the pre-defined models. The V2 key-pair models are based on specific certificate types, which have certificate definitions to correspond to each key function included in the certificate type. Each certificate definition has, by default, an associated certificate definition policy that contains policy settings tailored appropriately for that certificate definition.

[Table 13 on page 123](#) shows the characteristics of each model, listing the default certificate type, certificate definitions, and associated certificate definition policy for each model. If you need to look at the certificate definitions before you can decide whether the default is suitable, see [“Customizing certificates” on page 525](#). For a detailed description of each policy setting, see [“Certificate definition policy attributes reference” on page 432](#).

In most cases, the default values in the certificate definitions and policy settings are designed appropriately for the specific type of user. If your users fall into one of the default categories, see [“Default configuration path” on page 126](#) for instructions on configuring these key-pair users.

In some cases, the default values are not appropriate for your users, and you may need to change the certificate definition or policy settings. For example:

- If you have V2 3-key-pair EFS users who also require the Windows Smart Card Logon capability, you must create a new certificate type with a certificate definition that specifies the Smart Card Logon capability.

You must ensure that the certificate definition policy reflects this requirement (for information about configuring Smart Card Logon, see the Entrust white paper *Interoperating with Microsoft PKI-enabled applications*).

- If you have two groups of V2 4-key-pair users, where one group requires 2048-bit keys and the other group requires 1024-bit keys, you need to have two certificate definitions with different extensions.

If the default values do not suit your users, see [“Customized configuration path” on page 127](#) for instructions on configuring these users.

---

**Note:** In the following table, any 2-key-pair entry with encryption and verification certificates may use the V1 model if the client application is V1. For more information, see the *Security Manager Deployment Guide*.

---

**Table 13:** Predefined V2 key-pair certificate types

V2 model	Default certificate type	Certificate definitions	Associated certificate definition policy
1-key-pair	1 Key-Pair User (ent_skp_dualusage)	Dual Usage	Dual Usage Policy
1-key-pair	Enterprise Domain Controller (ent_ad_dc)	Dual Usage	Enterprise Domain Controller Policy
1-key-pair	Enterprise Machine (ent_machine)	Dual Usage	Enterprise Machine Policy
1-key-pair	ePassport - Document Signer (epass_doc_signer)	Document signer	Document Signer Policy
1-key-pair	MS VPN Client User (vpn_client_user)	Dual Usage	Dual Usage Policy
1-key-pair	Smart Card Logon for MS Security Framework Users (ent_ms_smrtcrd_capi)	Dual Usage	Dual Usage No Key Backup Policy
2-key-pair	2-Key-Pair User (ent_twokeypair)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• Encryption Policy</li> <li>• Verification Policy</li> </ul>
2-key-pair	Admin Services User Management (ent_admsrvcs_usrmgmt)	<ul style="list-style-type: none"> <li>• Verification</li> <li>• Encryption</li> </ul>	<ul style="list-style-type: none"> <li>• TruePass Server Verification Policy</li> <li>• none</li> </ul>

**Table 13:** Predefined V2 key-pair certificate types (continued)

V2 model	Default certificate type	Certificate definitions	Associated certificate definition policy
2-key-pair	Admin Services User Registration (ent_admsrvcs_userreg)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	Default (ent_default)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	Desktop Admin (ent_desktop)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	Email Content Scanner (ent_msgscanner)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	ePassport - IS Attached Client (ent_eacattached)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	ePassport - IS Concentrator (ent_eacon)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	ePassport - IS Standalone Client (ent_eacStandalone)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	ePassport - Master List Signer (ent_mlist_signer)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• Encryption Policy</li> <li>• Master List Signer Policy</li> </ul>
2-key-pair	ePassport - Master List Signer Administrator (ent_mlist_admin)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	ePassport - SPOC Administrator (ent_spoc_admin)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	ePassport - SPOC Client (ent_spoc_client)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>

**Table 13:** Predefined V2 key-pair certificate types (continued)

V2 model	Default certificate type	Certificate definitions	Associated certificate definition policy
2-key-pair	ePassport - SPOC DV Client (ent_spoc_dv)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	ePassport - SPOC Server (ent_spoc_server)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	Export (ent_export)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	IPSec Device (vpn_dir)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	IPSec Device (vpn_nodir)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	Messaging Server (ent_msgserver)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	PKCS10 2-Key-Pair User (ent_twokeypair_p10)	<ul style="list-style-type: none"> <li>• Encryption_p10</li> <li>• Verification_p10</li> </ul>	<ul style="list-style-type: none"> <li>• Encryption_p10 Policy</li> <li>• Verification_p10 Policy</li> </ul>
2-key-pair	Roaming Server (ent_profsrvr)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	Smart Card Logon for PKCS#11 Users (ent_msft_smartcard)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	Standalone EFS User (ent_standalone_efs)	<ul style="list-style-type: none"> <li>• EFS</li> <li>• CMP Signing</li> </ul>	<ul style="list-style-type: none"> <li>• MS EFS Policy</li> <li>• MS CMP Signing Policy</li> </ul>
2-key-pair	Timestamping Agent (ent_timestamp)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	Timestamping Agent Critical (ent_timestamping)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
2-key-pair	TruePass Server (ent_truepass)	<ul style="list-style-type: none"> <li>• Verification</li> <li>• Encryption</li> </ul>	<ul style="list-style-type: none"> <li>• TruePass Server Verification Policy</li> <li>• none</li> </ul>

**Table 13:** Predefined V2 key-pair certificate types (continued)

V2 model	Default certificate type	Certificate definitions	Associated certificate definition policy
2-key-pair	TruePass Server Multidomain Primary (ent_truepass_multi)	<ul style="list-style-type: none"> <li>• Verification</li> <li>• Encryption</li> </ul>	<ul style="list-style-type: none"> <li>• TruePass Server Verification Policy</li> <li>• none</li> </ul>
2-key-pair	XAP Server (ent_xapsrv)	<ul style="list-style-type: none"> <li>• Encryption</li> <li>• Verification</li> </ul>	<ul style="list-style-type: none"> <li>• none</li> <li>• none</li> </ul>
3-key-pair	EFS User (ent_efs)	<ul style="list-style-type: none"> <li>• EFS encryption</li> <li>• Verification</li> <li>• Encryption</li> </ul>	<ul style="list-style-type: none"> <li>• EFS Policy</li> <li>• Verification Policy</li> <li>• Encryption Policy</li> </ul>
3-key-pair	Nonrepudiation User (ent_nonrepud)	<ul style="list-style-type: none"> <li>• Nonrepudiation</li> <li>• Verification</li> <li>• Encryption</li> </ul>	<ul style="list-style-type: none"> <li>• Nonrepudiation Policy</li> <li>• Verification Policy</li> <li>• Encryption Policy</li> </ul>
4-key-pair	NonRepudiation/EFS User (ent_nonrepud_and_efs)	<ul style="list-style-type: none"> <li>• Verification</li> <li>• Nonrepudiation</li> <li>• EFS</li> <li>• Encryption</li> </ul>	<ul style="list-style-type: none"> <li>• Verification Policy</li> <li>• Nonrepudiation Policy</li> <li>• EFS Policy</li> <li>• Encryption Policy</li> </ul>

## Default configuration path

If you take the default path, the configuration process includes the following steps:

- 1 Select a certificate type and certificate definition policy.

The selection of a certificate type is essentially made when you determine which key pairs you need (see [“Supported key-pair models” on page 113](#)). In the default cases, the certificate type dictates the certificate definitions and their associated policy certificates, as shown in [Table 13 on page 123](#).

- 2 Creating the user and assign the certificate type.

To create the user, see [“Creating new users” on page 146](#), and select the appropriate certificate type on the **Certificate Info** property page.

---

**Attention:** If you are creating a V2-key-pair user, you cannot create the profile in Security Manager Administration. A V2-enabled client application must create the user's profile.

---

## Customized configuration path

If your users do not fit into the default stream, you must do some or all of the following tasks:

- 1 Customize a certificate type.

Each certificate type and certificate definition is specified in the `master.certspec` file. For information about editing the `master.certspec` file, see [“Customizing certificates” on page 525](#).

- 2 Customize a certificate definition policy.

In special cases, you may want to change a user policy or create a new user policy. See [“Administering user policies” on page 391](#) for more information.

- 3 Map a user policy to a certificate definition.

If you have created a new certificate definition or a new policy, you need to map the policy to the appropriate certificate definition. See [“Mapping policy certificates to certificate definitions” on page 404](#) for details.

- 4 Create the user and assign the certificate type.

To create the user, see [“Creating new users” on page 146](#), and select the appropriate certificate type on the **Certificate Info** property page.

---

**Attention:** If you are creating a V2-key-pair user, you cannot create the profile in Security Manager Administration. A V2-enabled client application must create the user's profile.

---

Typically, there are two different configuration scenarios, depending on whether you have different groups of users with different needs:

- If you have only one group of users, you can customize their configuration by editing either the default certificate definition or the certificate definition policy.
- If you have more than one group, you must create a new, customized certificate definition or user policy (or both) for each group, and then map the associations between the new certificate definitions and policies.





## Administering users

A user is any entry in the Security Manager database or directory. Users can be actual end users or Entrust PKI administrators in your organization, or non-human entries such as Web servers or other hardware devices.

The most common operations that Entrust PKI administrators perform are operations related to users. Entrust PKI administrators with sufficient permissions can add users to Security Manager, modify their properties, and manage their certificates. The operations that Entrust PKI administrators can perform on users depends on the administrator's role. For more information about roles, see [“Administering roles” on page 353](#).

This chapter contains the following sections:

- [“User states” on page 131](#)
- [“Email addresses in Security Manager” on page 132](#)
- [“Using special characters in user names” on page 133](#)
- [“Finding users” on page 134](#)
- [“Creating new users” on page 146](#)
- [“Adding existing users” on page 152](#)
- [“Managing activation codes” on page 153](#)
- [“Creating user profiles” on page 158](#)
- [“Recovering user key pairs” on page 162](#)
- [“Recovering user profiles” on page 165](#)
- [“Restricting users” on page 169](#)
- [“Deactivating and reactivating users” on page 172](#)
- [“Revoking user certificates” on page 174](#)
- [“Suspending user certificates” on page 181](#)
- [“Archiving and retrieving users” on page 183](#)

- “Removing users from the database” on page 191
- “Modifying distinguished names” on page 192
- “Restoring user certificates to the directory” on page 201
- “Moving users to a new Certification Authority” on page 202
- “Configuring user properties” on page 219
- “Updating key pairs” on page 240
- “Notifying client applications” on page 242
- “Changing user profiles” on page 243
- “Allowing profile export” on page 244
- “Converting V2 users to V1 users” on page 247

# User states

Users will be in various states throughout their lifecycle. Table 14 describes the different states. The Entrust state information does not necessarily correlate with any directory state information. For example, you can delete users with your directory tools, the users and their certificates still exist in the database as valid. You must manually update Security Manager to synchronize the changes.

**Table 14:** User states in Security Manager Administration

State	Description
Non-Entrust	The user exists in the Security Manager directory, but is not added to Security Manager or the Security Manager database yet (the user does not have an Entrust profile or certificates).
Added	The user was added to the Security Manager database, but no profile is created yet. If a user entry did not exist in the directory, the entry was created. If a directory entry already existed, the entry was updated with Security Manager attributes.
Active	The user exists in the directory and the database, and a profile was created. The user can log in to a Security Manager client with the profile.
DN Change	The user is configured to complete a distinguished name (DN) change the next time the user logs in to a Security Manager client. The user entry is changed in the directory, but requires an update in the database.
Deactivated	An administrator deactivated the user. The user exists in both the directory and database.
Key Recovery	An administrator has set the user for a key recovery. The user exists in both the directory and database.
Export Hold	The user is being exported to a new Certification Authority (CA), but the user has not yet been imported to the new CA. The user exists in both the directory and database.
Export	The user was exported to a new CA. An administrator for the old CA must manually change the state from Export Hold to Export. The user exists in both the directory and database.
Import Key Recovery	The user was imported into a new CA. After the user logs in to a Security Manager client, the state automatically changes to Active. The user exists in both the directory and database.
Archived	The user is no longer an Entrust user. The user exists in database, but may or may not exist in the directory. <b>Note:</b> You can recover archived users.

# Email addresses in Security Manager

If you are using S/MIME applications (such as Entrust Entelligence Security Provider for Windows), you need to include the email address of the user in the `subjectAltName` extension.

By default, the Person user type (see [“Overview of user template file and user types” on page 448](#)) allows you to add one email address for each user entry you create. This email address is usually stored in the directory using the `mail` attribute.

By default, Security Manager also auto-populates the values in the `mail` attribute into the `subjectAltName` extension in the user's certificates. If you are using an attribute other than `mail`, you must update the user template file.

By default, you can add multiple email addresses when creating a user. This process ensures that all email addresses are stored in the directory and the `subjectAltName` component values when the user is created. You also have the following options:

- After creating the user, modify the `subjectAltName` component values by adding the additional email address(es).  
See [“Adding, modifying, or deleting subjectAltName component values” on page 260](#). This process does not update your directory. Use this option if you are using Active Directory and you cannot add multiple values to the same attribute.
- After creating the user, update the directory using the Directory Browser. See [“Using the Directory Browser” on page 69](#).  
Then use Security Manager Administration to update the user entry based on your directory changes. See [“Updating subjectAltName component values from the directory” on page 263](#).

If the user entry already exists in the directory and the email attribute already contains multiple email addresses, Security Manager automatically adds all addresses to the `subjectAltName` component when the user is created.

## Including email addresses in distinguished names

Due to changing recommendations in the current S/MIME standard, it is recommended that you do not include an email address in a user's DN. Rather, include the email address in the `subjectAltName` attribute of the user's X.509 certificate.

Existing users who already have their email address in their DN may continue to work with compliant S/MIME applications. However, you should update such users so their email address is moved into the `subjectAltName`. To do this, change the users' DNs (see [“Changing distinguished names” on page 193](#)).

# Using special characters in user names

When you add a user with a name that includes special characters, consider the following:

- Type the characters just as they would be represented in the directory.
- You can add the following characters in attributes that make up the DN without adding quotation marks: # + ; = . \
- Characters that you cannot use in the DN include the vertical bar (|) and square brackets ([]).

---

**Note:** You may use slashes in the DN or attribute of a user. If you use the forward (/) or backward (\) slash character, precede it with a backward slash. A single slash will appear in the certificate and directory. For example, if you enter \ /, only the forward slash (/) will appear.

---

Contact a Directory Administrator for more information about including special characters in names when you add users.

- To include quotation marks (") in an attribute that goes into the DN, type a backslash before each quotation mark.
- To include the short form of a user's name, for example, "Drew" for Andrew, type the following in the **First Name** field:

Andrew \ "Drew"

---

**Note:** The backslashes still appear in the DN, but are omitted in the directory (the CN appears as Andrew "Drew").

---

- To include a comma in the DN, you must precede the comma with a backslash or enclose the entry in double quotes. For example:

Gray\, Alice

or

"Gray, Alice"

- To include a forward slash in the DN, you must precede the forward slash with a backslash. For example, to include the serial number "RSH-21/12", type

RSH-21\12

# Finding users

You can find users in both the Security Manager database and the directory, based on database or directory information. Searching for users by Entrust properties (such as distinguished name, role, or group) finds users that reside in the Security Manager database. Searching for users by directory attributes (such as name or serial number) finds users that reside in the directory.

---

**Note:** You can also find users in the directory using the Directory Browser (see [“Finding entries in the directory” on page 72](#)).

---

This section contains the following topics:

- [“Finding users by Entrust properties” on page 134](#)
- [“Finding users by directory attributes” on page 142](#)

## Finding users by Entrust properties

Searching for users by Entrust properties (such as distinguished name, role, or group) finds users that reside in the Security Manager database.

For details about configuring how Security Manager Administration performs searches, see [“Configuring search performance preferences” on page 60](#).

After generating a user list you can save the user list to a text file. You can view saved user list text files using any text editor. Since the text file is tab delimited, you can also import it into a spreadsheet program for easier viewing.

---

**Note:** To decrease the amount of time it takes to find entries, specify narrow criteria for your search.

---

### To find users by Entrust properties

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Select **Users > Find > By Entrust Properties**.  
The **Find Users by Entrust Properties** dialog box appears.

- 3 To revert all search options back to the defaults, click **Reset**.
- 4 Click the **General** tab.

---

**Note:** All DN's, group names, role names, and revocation comments are converted to the UTF-8 character set before being placed in the search results.

---

- In the **DN** field, enter the distinguished name (DN) of the user you want to find.  
 When you search for entries with directory attribute values that include special characters, the values you enter must match the directory entry.  
 Optionally, you can use wildcards. Wildcards let you search for partial attributes. Add an asterisk (\*) with a partial search string to find users that include the search string information. For example, enter \*dr\* to find all entries named Andrew and Drew. Enter dr\* to exclude Andrew and find only entries beginning with the letters dr.
- In the **Current State** drop-down list, select a user state to find users in a specific state, or select **<Any>** to include all user states. For details about user states, see ["User states" on page 131](#).

- In the **Role** drop-down list, select a role to find users with a specific role, or select **<Any>** to find users with any role. For more information about roles, see [“Administering roles” on page 353](#).
- In the **Group** drop-down list, select a group to find users who belong to that specific group, **All groups** to find users who belong to all groups, or **<Any>** to find users who belong to any group. For more information about groups, see [“Administering groups” on page 329](#).
- In the **Attribute Certificates** drop-down list, select **<With>** to find users with attribute certificates, **<Without>** to find users without attribute certificates, or **<Any>** to find users with or without attribute certificates. For more information about attribute certificates, see [“Administering attribute certificates” on page 635](#).
- In the **Activation Codes** drop-down list, select **<With>** to find users with activation codes, **<Expired>** to find users with expired activation codes, or **<Any>** to find users with or without activation codes. For more information about activation codes, see [“Managing activation codes” on page 153](#).

**5** Click the **Certificates** tab.

The screenshot shows a dialog box titled "Find Users by Entrust Properties" with a close button (X) in the top right corner. The dialog has four tabs: "General", "Certificates", "State History", and "Range". The "Certificates" tab is currently selected. Inside the dialog, there are several fields and a checkbox:

- Certificate Category:** A dropdown menu with "<Any>" selected.
- Certificate Type:** A text box containing "< Any >".
- Revoked:** A dropdown menu with "< Any >" selected.
- Expired:** A dropdown menu with "< Any >" selected.
- Signing Usage Ends:** A dropdown menu with "< Any >" selected.
- Issued:** A dropdown menu with "< Any >" selected.
- Policy:** A dropdown menu with "< Any >" selected.
- Key Update Pending:** A checkbox that is currently unchecked.

At the bottom of the dialog, there are four buttons: "OK", "Cancel", "Reset", and "Help".



---

**Note:** Using the **Revoked**, **Expired**, **Signed Expired**, or **Issued** drop-down lists, you can specify the latest certificate as part of your search criteria. However, Security Manager can distinguish the latest certificate only if Security Manager 7.0 or later generated the certificate. Certificates created by an earlier version of Security Manager are not considered a latest certificate and are not included in the search results.

---

- In the **Certificate Category** field, select a certificate category to find users with certificates from a specific certificate category, or select **<Any>** for any certificate category.

If you selected a certificate category, the **Certificate Type** field becomes available.

- In the **Certificate Type** field, select a certificate type to search for users with a specific certificate type, or select **<Any>** to find users with any certificate type for the certificate category.
- In the **Revoked** drop-down list, select:
  - **<Any>** to find users with any certificates during the date range, regardless of revoked status
  - **<All revoked>** to find users with all their certificates revoked during the date range
  - **<Some revoked>** to find users with at least one revoked certificate during the date range
  - **<None revoked>** to find users with no revoked certificates during the date range
  - **<Latest revoked>** to find users with any of their latest certificates revoked during the date range  
The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).
  - **<Latest not revoked>** to find users with none of their latest certificates revoked during the date range  
The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).
- In the **Expired** drop-down list, select:
  - **<Any>** to find users with any certificates during the date range, regardless of expired status
  - **<All expired>** to find users with all their certificates expired during the date range
  - **<Some expired>** to find users with at least one expired certificate during the date range

- **<None expired>** to find users with no expired certificates during the date range
- **<Latest expired>** to find users with any of their latest certificates expired during the date range  
The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).
- **<Latest expired in current stream>** to find users with any of their latest certificates expired in the current stream during the date range  
The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).  
A stream for a user is identified by a certificate type and certificate definition. A user can have more than one stream if the user ever changed certificate types. The current stream is identified by the user's current certificate type.
- **<Latest not expired>** to find users with none of their latest certificates expired during the date range  
The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).
- In the **Signing Usage Ends** drop-down list, select:
  - **<Any>** to find users with any private signing keys, regardless of expired status
  - **<All expired>** to find users with all their private signing keys expired
  - **<Some expired>** to find users with at least one of their private signing keys expired during the date range
  - **<None expired>** to find users with no expired certificates
  - **<Latest expired>** to find users with any of their latest private signing keys expired  
The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).
  - **<Latest expired in current stream>** to find users with any of their latest private signing keys expired in the current stream during the date range  
The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).  
A stream for a user is identified by a certificate type and certificate definition. A user can have more than one stream if the user ever changed certificate types. The current stream is identified by the user's current certificate type.

- **<Latest not expired>** to find users with none of their latest private signing keys expired

The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).

- In the **Issued** drop-down list, select:
  - **<Any>** to find all users with any certificates during the date range, regardless of issued status
  - **<All issued>** to find all users with all their certificates issued during the date range
  - **<Some issued>** to find all users with at least one issued certificate during the date range
  - **<None issued>** to find all users with no issued certificates during the date range
  - **<Latest issued>** to find all users with any of their latest certificates issued during the date range
 

The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).
  - **<Latest not issued>** to find all users with none of their latest certificates issued during the date range
 

The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).

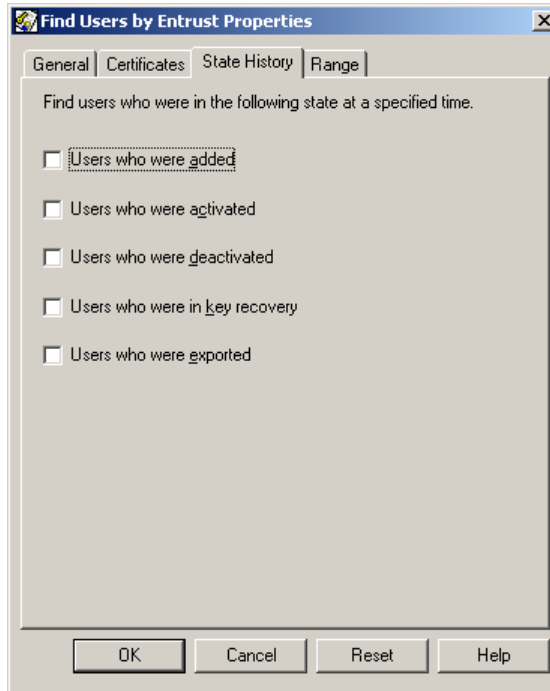
---

**Note:** The time the certificate is issued is not the same as the time the user is activated. The certificate issue time is set to 30 minutes after the user is activated.

---

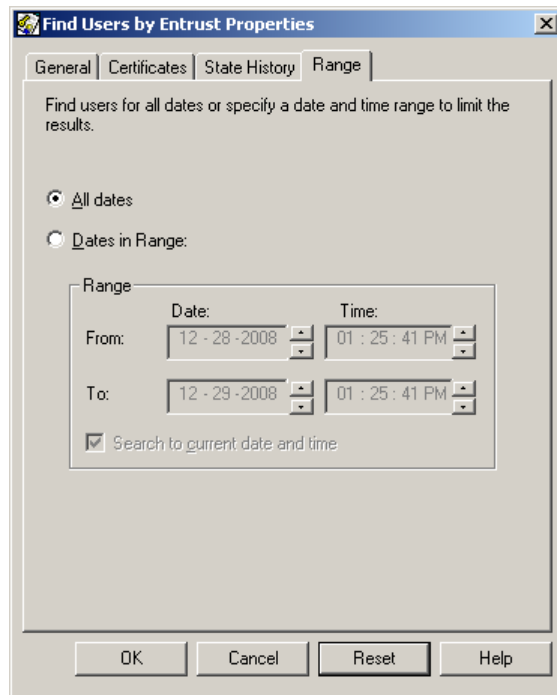
- In the **Policy** drop-down list, select:
  - **<Default lifetime>** to find all users with policy certificates set to the default lifetime.
  - **<Specified lifetime>** to find all users with policy certificates set to a specified lifetime.
  - **<Specified expiry date>** to find all users with policy certificates set to expire on a specific date.
  - **<Any>** to find all users with policy certificates, regardless of lifetime.
- To find users with a pending key update, select **Key Update Pending**.

**6** Click the **State History** tab.



- To find users who were in the Added state during the date range, select **Users who were added**.
- To find users who were in the Active state during the date range, select **Users who were activated**.
- To find users who were in the Deactivated state during the date range, select **Users who were deactivated**.
- To find users who were in the Key Recovery state during the date range, select **Users who were in key recovery**.
- To find users who were in the Export state during the date range, select **Users who were exported**.

**7** Click the **Range** tab.




---

**Note:** You can specify dates in the future. For example, you might want to find users whose certificates will expire in the next week.

---

- To search for users , click **All dates**.
- To specify a date range, click **Dates in Range** and then
  - Use the **From** combo boxes to enter or select a starting date and time.
  - To use the current date and time as the end of the date range, select **Search to current date and time**.
  - To specify a custom end date and time other than the current date and time, deselect **Search to current date and time** and then use the combo boxes to enter or select an ending date and time.

**8** Click **OK**.

A dialog box appears stating that the operation is successful. It also lists the number of users found that meet your search criteria.

**9** Click **OK**.

A list of all users that match the search criteria appear in the right pane of the Security Manager Administration window. By default, users are sorted

alphabetically by their distinguished name. You can change the sort order by clicking on one of the following columns:

- The **Distinguished Name (DN)** column lists the DNs of each user.
- The **State** column lists the state of each user in Security Manager. For a description of the user states, see [“User states” on page 131](#).
- The **Role** column list the role of each user.  
If the user has not been added to Security Manager, the user’s role is listed as **Undefined**.
- The **Group** column lists the groups assigned to each user.  
If the user has not been added to Security Manager, the user’s group list is listed as **Undefined**.
- The **Category** column lists the certificate category for each user.  
If the user has not been added to Security Manager, the user’s certificate category is listed as **Undefined**.
- The **Certificate Type** column lists the certificate type for each user.  
If the user has not been added to Security Manager, no certificate type is listed for the user.
- The **Protocol Version** column lists the Entrust digital ID type (V1 or V2) for each user.  
If the user does not have a digital ID, the digital ID type is listed as **Unassigned**.
- The **Attribute Certificates** column lists the labels of any attribute certificates issued to each user.  
If a user does not have any attribute certificates, no information is listed for the user.

Depending on your role permissions, not all columns may appear. For example, the **Attribute Certificates** column will not appear if you do not have the appropriate permissions.

**10** (Optional.) To save the results to a text file, select **Users > Save As**.

Your Security Manager Administration preferences determine the information that is included in the text file. See [“Configuring user list preferences” on page 62](#) for details.

## Finding users by directory attributes

Searching for users by directory attributes (such as name or serial number) finds users that reside in the directory. The ability to search for directory entries is an excellent way of adding users in the directory. For example, if your organization deploys Security Manager and wants to use an existing directory, you do not have to add each

user one at a time. You can simplify the process by finding a non-Entrust entry and adding the user in a single step.

The directory attributes you can find depend on the user type and the attributes that Entrust PKI administrators include for every user added.

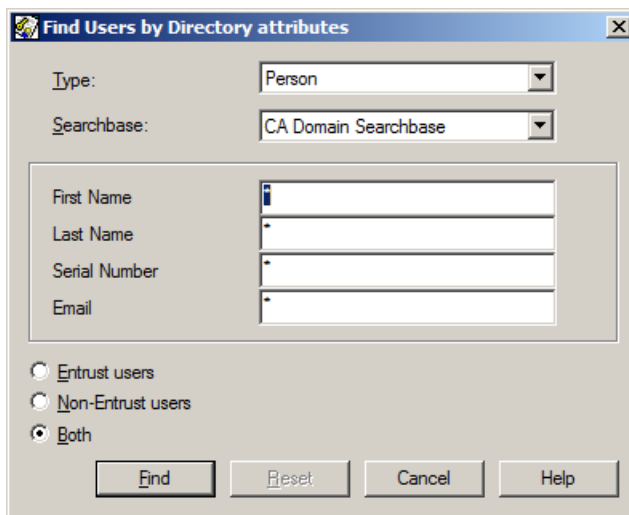
For details about configuring how Security Manager Administration performs searches, see [“Configuring directory search preferences” on page 59](#) and [“Configuring search performance preferences” on page 60](#).

After generating a user list you can save the user list to a text file. You can view saved user list text files using any text editor. Since the text file is tab delimited, you can also import it into a spreadsheet program for easier viewing.

### To find users by directory attributes

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Select **Users > Find > By Directory Attributes**.

The **Find Users by Directory attributes** dialog box appears.



- 3 To revert all options back to the defaults, click **Reset**.
- 4 In the **Type** drop-down list, select a user type.  
The type that you select determines which attribute fields appear. For example, if you select **Person** the **First Name**, **Last Name**, **Serial Number**, and **Email** fields appear. If you select **Organizational Unit**, only the **Organizational Unit** field appears.
- 5 Enter information in the attribute fields to define your search.

To decrease the amount of time it takes to find entries, specify narrow criteria for your search.

Optionally, you can use wildcards. Wildcards let you search for partial attributes. Add an asterisk (\*) with a partial search string to find users that include the search string information. For example, enter \*dr\* to find all entries named Andrew and Drew. Enter dr\* to exclude Andrew and find only entries beginning with the letters dr.

- 6 In the **Searchbase** drop-down list, select the searchbase containing the users that you want to find. For more information about searchbases, see [“Administering searchbases” on page 341](#).
- 7 Select one of the following options.

---

**Note:** The following options do not appear if you use Microsoft Active Directory.

---

- To find users who are added, deactivated, active, set up for key recovery, or set up for a DN change, click **Entrust PKI users**.
- To find users who were never added to Security Manager or users that were deactivated before they were activated, click **Non-Entrust PKI users**.
- To find all users in the searchbase, click **Both**.

- 8 Click **Find**.

A dialog box appears stating that the operation is successful.

- 9 Click **OK**.

A list of all users that match the search criteria appear in the right pane of the Security Manager Administration window. By default, users are sorted alphabetically by their distinguished name. You can change the sort order by clicking on one of the following columns:

- The **Distinguished Name (DN)** column lists the DNs of each user.
- The **State** column lists the state of each user in Security Manager. For a description of the user states, see [“User states” on page 131](#).
- The **Role** column list the role of each user.  
If the user has not been added to Security Manager, the user’s role is listed as **Undefined**.
- The **Group** column lists the groups assigned to each user.  
If the user has not been added to Security Manager, the user’s group list is listed as **Undefined**.
- The **Category** column lists the certificate category for each user.  
If the user has not been added to Security Manager, the user’s certificate category is listed as **Undefined**.



- The **Certificate Type** column lists the certificate type for each user.  
If the user has not been added to Security Manager, no certificate type is listed for the user.
- The **Protocol Version** column lists the Entrust digital ID type (V1 or V2) for each user.  
If the user does not have a digital ID, the digital ID type is listed as **Unassigned**.
- The **Attribute Certificates** column lists the labels of any attribute certificates issued to each user.  
If a user does not have any attribute certificates, no information is listed for the user.

Depending on your role permissions, not all columns may appear. For example, the **Attribute Certificates** column will not appear if you do not have the appropriate permissions.

**10** (Optional.) To save the results to a text file, select **Users > Save As**.

Your Security Manager Administration preferences determine the information that is included in the text file. See [“Configuring user list preferences” on page 62](#) for details.

# Creating new users

Creating users is one way of adding users to Security Manager. When you create a new user, the user is added to the Security Manager directory and database. Create new users if the user does not exist in the directory. If the user already exists in the directory, you must add the user to Security Manager (see [“Adding existing users” on page 152](#)).

You cannot create new users if you use Microsoft Active Directory. For Microsoft Active Directory, you must first add the user to the directory using your directory tools and then add the user to Security Manager.

Before entering user names or email addresses, read [“Email addresses in Security Manager” on page 132](#) and [“Using special characters in user names” on page 133](#) for important information.

## To create a new user

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Select **Users > New User**.

The **New User** dialog box appears. An asterisk (\*) appears beside required fields. You must enter information into all fields marked with asterisks.

The information required under the **Naming** tab may differ from that described in this procedure if a Security Officer or Entrust PKI administrator has modified the template definition file. Contact a Security Officer if you have any questions about the appearance of this dialog box.

**New User**

Naming | General | Certificate Info | Key Update Options

Type: **Person**

This is the user type to be used for most Entrust users.

In DN

• First Name		<input checked="" type="checkbox"/>
• Last Name		<input type="checkbox"/>
Serial Number		<input type="checkbox"/>
Email		<input type="checkbox"/>

Asterisks (\*) appear beside required attributes.

Add to: **CA Domain Searchbase**

Show DN...

☐ Create profile

OK Cancel Help

### 3 Click the **Naming** tab.

The information required under the **Naming** tab may differ from that described in this procedure if an Entrust PKI administrator has modified the template definition file. Contact a Security Officer if you have any questions about the appearance of this dialog box.

#### a In the **Type** drop-down list, select a user type.

The type that you select determines which attribute fields appear. For example, if you select **Person**, the **First Name**, **Last Name**, **Serial Number**, and **Email** fields appear. If you select **Web server**, the **Name** and **Description** fields appears.

Typically, you select **Person** to add end users and other administrators to Security Manager. You typically select **Web server** to add entities that require Web certificates, such as Web servers or VPN servers.

---

**Note:** When you add a user with a name that includes special characters, ensure that you enter the characters just as they are represented in the directory. For more information about using special characters in names, see [“Using special characters in user names” on page 133](#).

To add an attribute to the DN that is not western European (for example, Japanese), see [“Entering international characters in distinguished names” on page 679](#).

---

- b** Enter information into the available attribute fields. An asterisk (\*) appears beside required fields. You must enter information into all fields marked with asterisks.

For example, if you select **Person** as the user type:

- In the **First Name** field, enter the user’s first name.
- In the **Last Name** field, enter the user’s last name.
- (Optional.) In the **Serial Number** field, enter the user’s serial number (for example, the employee number).

Depending on your organization, the serial number may not be the employee number. Check with a Security Officer if you are unsure which number to enter.

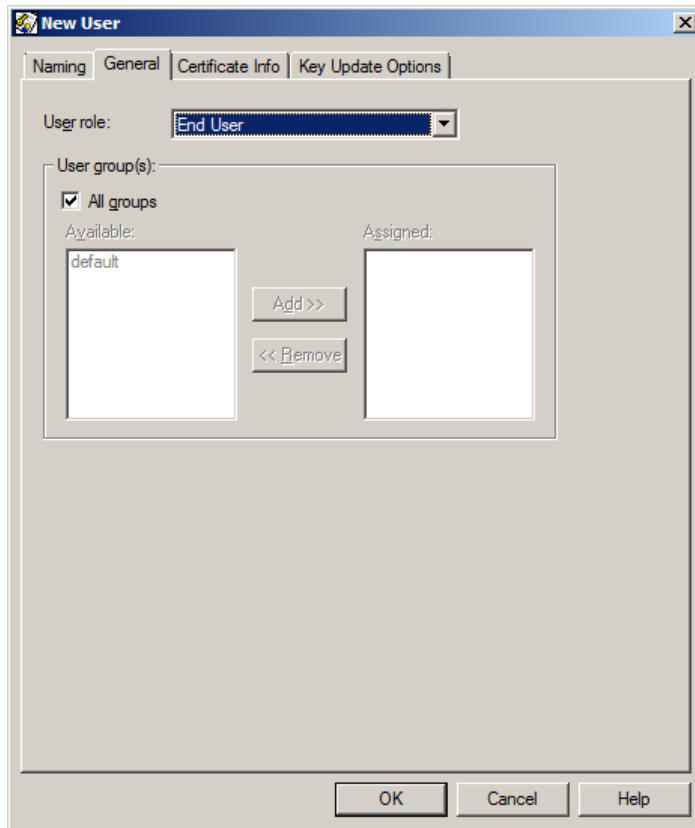
- (Optional.) In the **Email** field, enter the user’s email address. To enter more than one email address, separate each email address multiple with a space. For more information, see [“Email addresses in Security Manager” on page 132](#).

- c** In the **Add to** drop-down list, select the searchbase where you want to add the user. For more information about searchbases, see [“Administering searchbases” on page 341](#).

- d** To create a digital ID for the user, select **Create profile**.

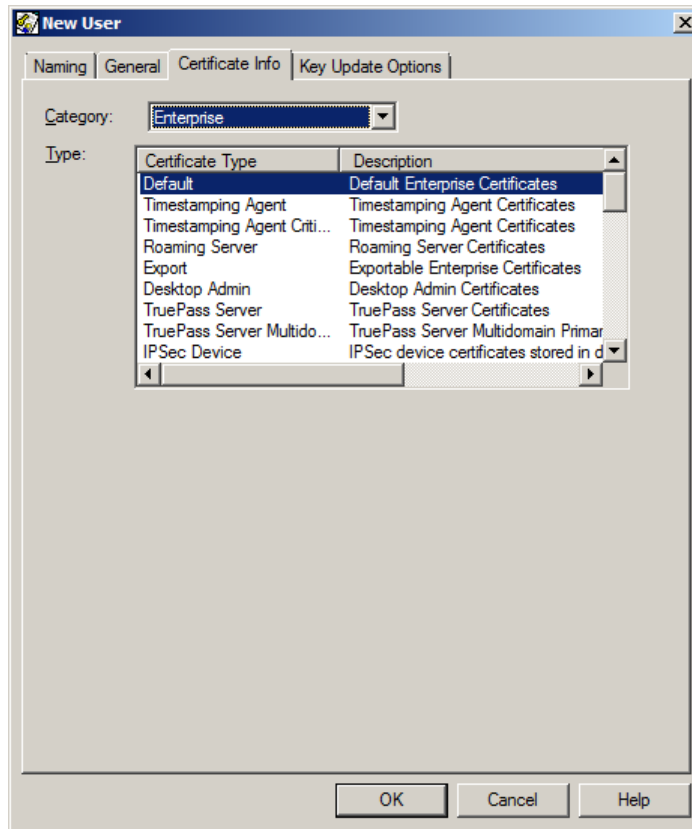
Typically, you select this option if you are creating digital IDs for Security Manager client applications. If you select this option, Security Manager Administration prompts you for a location, name, and password for the digital ID and Entrust support files. Otherwise, Security Manager generates activation codes (reference number and authorization code). Users enter the activation codes into their client application to generate their digital ID.

- 4** Click the **General** tab.



- a** In the **User role** drop-down list, select a role for the user. For more information about roles, see [“Administering roles” on page 353](#).
- b** Under **User group(s)**:
  - To add the user to all groups, select **All groups**. (This adds the user to all current and future groups.)
  - To add the user to specific groups, deselect **All groups** and add or remove groups as required.  
 To add a group, select a group from the **Available** list and then click **Add**.  
 To remove a group, select a group from the **Assigned** list and then click **Remove**.

**5** Click the **Certificate Info** tab.



- a In the **Category** drop-down list, select a certificate category and then select a certificate type from the **Type** list. For more information about certificates, see ["Customizing certificates" on page 525](#).
- b If **Certificate Extension** fields are displayed, enter the certificate extension variables into the **Certificate Extension** fields.  
If you do not have this information, or need additional information, contact a Security Officer. Extension variables are defined in the `master.certspec` file. For information about this file, see ["Customizing certificates" on page 525](#).
- 6 Click the **Key Update Options** tab. The information under this tab allows you to set key lifetimes and key expiry dates. Most organizations use the default values. To set key lifetimes other than those selected by default, see ["Configuring user key update options" on page 230](#).
- 7 Click **OK** to create the user.

- 8 If prompted, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).
- 9 If you chose to create a profile for the user, Security Manager Administration prompts you to provide the information necessary to create the profile. See [“Creating user profiles” on page 158](#) for information about creating a profile for the user.

# Adding existing users

If users already exist in the directory, you can add them to Security Manager. For details about creating new users, see [“Creating new users” on page 146](#).

---

**Note:** If you modify the user template file to create your own user type, the user's directory entry must match that user type when adding the user. For example, all of the object classes defined in the template type must be present in the directory entry. See [“Modifying the user template file and user types” on page 447](#).

---

## To add a user from the directory

- 1 Log in to Security Manager Administration. See [“Logging in to Security Manager Administration” on page 46](#).
- 2 Find users by directory attributes (see [“Finding users by directory attributes” on page 142](#)).
- 3 Select the user you want to add to Security Manager.
- 4 To add the user to Security Manager with default properties, select **Users > Selected User > Add to Entrust**.
- 5 To add the user to Security Manager with custom properties, select **Users > Selected User > Properties** and then modify the user properties as described in [“Configuring user properties” on page 219](#).
- 6 If prompted, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.



# Managing activation codes

Activation codes refer to the reference number and authorization code generated when Entrust PKI administrators add users to Security Manager, recover users' keys, or reissue activation codes.

Users need activation codes to create an Entrust digital ID (also called an Entrust profile) in their Security Manager client application. Users must create their digital ID before Security Manager can activate them.

Entrust PKI administrators with sufficient permissions are responsible for distributing activation codes to new users after adding them to Security Manager.

This section contains the following topics:

- [“Distributing activation codes” on page 153](#)
- [“Configuring the lifetime of activation codes” on page 154](#)
- [“Viewing activation codes and expiry dates” on page 155](#)
- [“Reissuing activation codes” on page 156](#)

## Distributing activation codes

In most organizations, Security Officers decide how to distribute activation codes. You can securely distribute activation codes:

- verbally in person or over the phone
- by email
- through a secure channel

For example, you can write them down, insert them into a sealed envelope and distribute through the internal corporate mail system.

- by writing them to a file

You should distribute the reference number and authorization code using different methods. For example, you could tell the user the reference number on the phone, and send the authorization code by email. Distributing them separately using different methods increases the security of the activation codes and helps prevent an attacker from obtaining access to the activation codes.

Alternatively, Entrust PKI administrators with sufficient permissions can create activated Entrust digital IDs. For more information, see [“Creating user profiles” on page 158](#).

## Configuring the lifetime of activation codes

To provide an additional layer of security in the handling of activation codes, you can limit the length of time that activation codes are valid—between 1 and 365 days. This is a global setting that affects all users who are activated or set for key recovery. You cannot override this feature on a per-user basis. By default, activation codes expire in 14 days. See [“Viewing activation codes and expiry dates” on page 155](#).

When setting the lifetime for activation codes, allow enough time for your organization to deliver the codes to users, and time for users to install their client application and create their digital ID. For example, a lifetime of five days allows for three days to deliver the activation codes and two days for the user to install the client application and create a digital ID.

---

**Note:** While restricting the lifetime of activation codes adds another level of security, guard activation codes so they are not lost or stolen during delivery.

---

If a user’s activation codes expire before the user can create a digital ID, you can reissue the activation codes. See [“Reissuing activation codes” on page 156](#) for details.

### To configure the lifetime of activation codes

- 1 Log in to Security Manager Administration. See [“Logging in to Security Manager Administration” on page 46](#).
- 2 In the tree view, select **Security Policy**.
- 3 Click the **Administration Policy** tab.
- 4 In the **Activation codes lifetime (in days)** field, enter the lifetime for activation codes (from 1 to 365)
- 5 Click **Apply**.
- 6 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

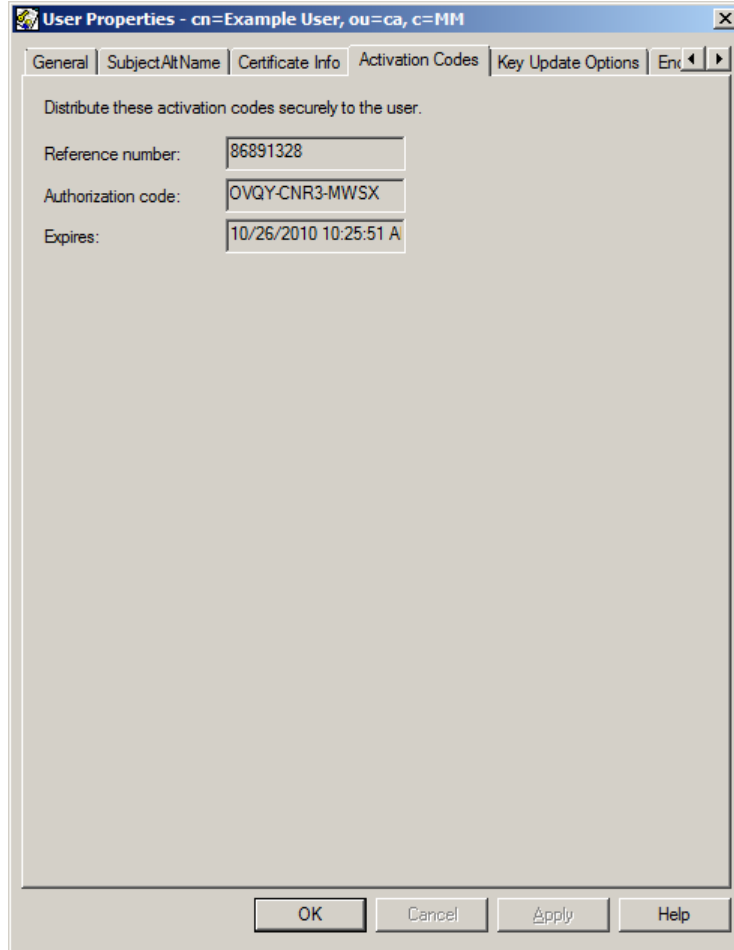
## Viewing activation codes and expiry dates

You can view the activation codes for users in the Added or Key Recovery states (see [“User states” on page 131](#)). You cannot view activation codes for activated or deactivated users. Activated users have already used their activation codes to create their profiles.

Expired activation codes are crossed out with Xs. When activation codes expire, you must reissue activation codes to the user (see [“Reissuing activation codes” on page 156](#)).

### To view activation codes and expiry dates

- 1** Log in to Security Manager Administration. See [“Logging in to Security Manager Administration” on page 46](#).
- 2** Find the user whose activation codes you want to view (see [“Finding users by directory attributes” on page 142](#)).
- 3** Select **Users > Selected User > Properties**.  
The **User Properties** dialog box appears.



**4** Click the **Activation Codes** tab.

You can now view the activation codes and expiry date. If required, copy the codes and send them securely to the user (see [“Distributing activation codes” on page 153](#)).

## Reissuing activation codes

Reissuing activation codes generates new codes for a user. You can reissue activation codes at any time. You do not have to wait until the old codes expire. You can reissue new codes to a user before the activation codes expire, for example, when codes are lost or stolen.

### To reissue activation codes

- 1** Log in to Security Manager Administration. See [“Logging in to Security Manager Administration” on page 46](#).
- 2** Find the user who requires new activation codes (see [“Finding users by directory attributes” on page 142](#)).
- 3** Select the user you want to add to Security Manager.
- 4** Select **Users > Selected User > Reissue Activation Codes**.
- 5** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears. New activation codes are generated and are available for viewing. See [“Viewing activation codes and expiry dates” on page 155](#).

# Creating user profiles

Typically, when you create or add users (see [“Creating new users” on page 146](#) and [“Adding existing users” on page 152](#)), Security Manager generates activation codes. Users enter these activation codes in their Security Manager client application to create a digital ID. Entrust PKI administrators can create digital IDs (also called profiles) for users.

---

**Note:** This feature is intended for non-human entities, such as Web servers or other hardware devices, or the profiles required to run Security Manager client applications. It is recommended that you do not create user profiles for users in your organization. Users should create their own profiles in their client application using their activation codes (see [“Managing activation codes” on page 153](#)).

---

You can create user profiles on software or on a hardware token.

- [“To create a user profile on software” on page 158](#)
- [“To create a user profile on a token” on page 159](#)

## To create a user profile on software

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Find the user whose requires a profile (see [“Finding users” on page 134](#)).
- 3** Select **Users > Selected User > Create Profile**.
- 4** If Security Manager Administration detects a hardware token, the **Select profile storage location** dialog box appears. Click **No** to store the profile on software. The **Create profile** dialog box appears.

- 5 To create a desktop profile, click **Create desktop profile** (selected by default). Otherwise, click **Create roaming profile** to create a roaming profile.  
You can only create a roaming profile if a Roaming Server is installed and running.
- 6 In the **Name** field, enter the name for the profile and Entrust support files.  
For example, if you enter `Security Officer`, the profile is named `Security Officer.epf`.
- 7 In the **Location** field, enter a location for the profile and Entrust support files, or click **Browse** to choose a location.
- 8 In the **Password** and **Confirm** fields, enter a password or have the user enter the password for you.  
The password must conform to the password rules. Click **Password Rules** to view the password rules.

---

**Note:** This password is very important. The user uses this password to access the activated profile. Record the password so you can give it to the user.

---

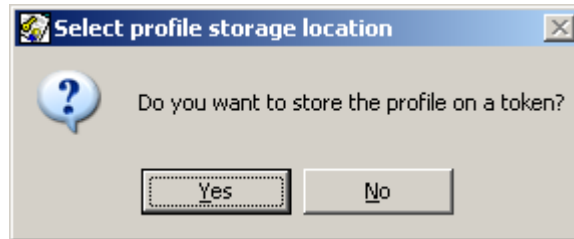
- 9 Click **OK** to create the profile.
- 10 Give the profile and password to the user.

#### To create a user profile on a token

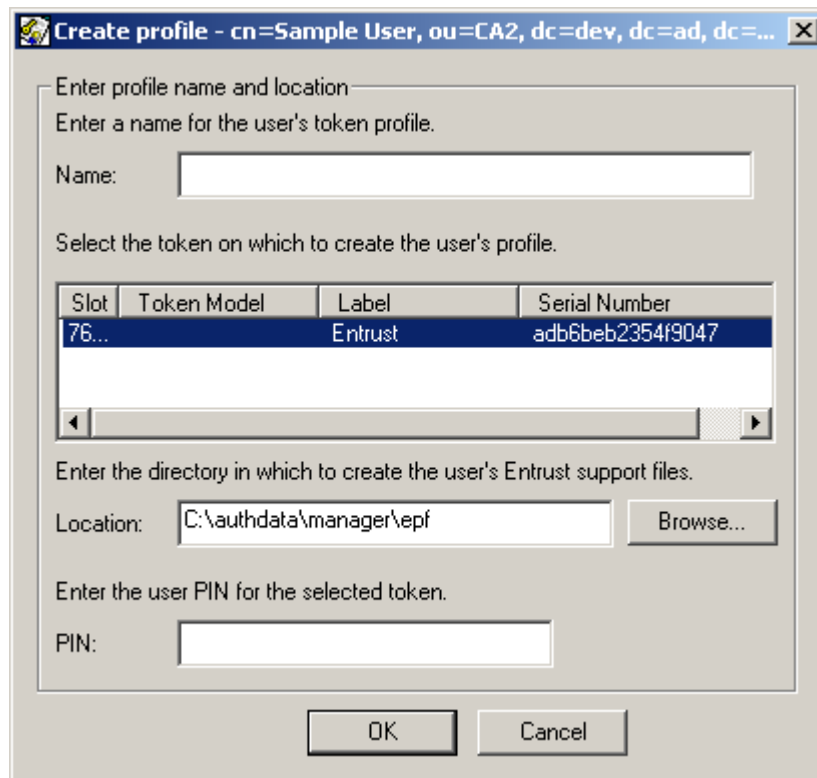
- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Ensure that the token that will contain the profile is in the token reader.

- 3 Find the user whose profile you want to create (see [“Finding users”](#) on page 134).
- 4 Select **Users > Selected User > Create Profile**.

The **Select profile storage location** dialog box appears.



- 5 Click **Yes** to store the profile to a token.
- The **Create Profile** dialog box appears.



- 6 In the **Name** field, enter the name for the profile and Entrust support files. Do not enter a name longer than 75 characters.
- 7 In the token list, select the token where you want to save the token.



- 8** In the **Location** field, enter a location for the Entrust support files, or click **Browse** to choose a location.
- 9** In the **PIN** field, enter the PIN previously assigned to the token.
- 10** Click **OK**.

If there is already a profile on the token, a message appears asking if you want to overwrite the existing profile. Click **No** to cancel the profile creation, or click **Yes** to overwrite the existing profile.

# Recovering user key pairs

Recover a user's keys when any of the following happens:

- When the user forgets a password. This is the most common occurrence.
- When a user loses their digital ID (profile) or the user's profile is damaged.
- When a user believes that the keys are compromised or that an attacker possesses the password or profile.

---

**Note:** Revoke a user's key if you suspect a key compromise. This ensures the generation of a new key pair. If you do not revoke the keys, other users will encrypt for the user with untrusted keys.

---

- When the user is set up to not have their key pairs automatically updated and their situation changes. For example, when a contractor's contract is extended and you need to issue new keys for the extension period.
- When the user's signing private key expires (which should rarely or never occur).

When you recover a user's keys, the user does not receive new encryption key pairs. Instead, Security Manager sends the user's client application a copy of the encryption key pair history.

Recovering a user's keys is a two-step process:

- 1** The Entrust PKI administrator changes the user state to Key Recovery. This generates new activation codes for the user.  
The Entrust PKI administrator gives these to the user. See ["Setting users for key recovery" on page 163](#).
- 2** The user enters the activation codes in the Security Manager client application and recreates a profile to complete the procedure. When a user completes the key recovery process, the client application creates a new signing key pair to replace the old one.

Alternatively, you can create a new profile for the user and give the user the profile and password instead of the activation codes. See ["Recovering user profiles" on page 165](#).

---

**Note:** When you recover a user's keys, the user's client application deletes the user's profile if it still exists. As a result, the user's password history, which is stored in the profile, is no longer available. This means that a recovered user can choose the same password for the new profile as the previous profile.

---

This section contains the following topics:

- ["Setting users for key recovery" on page 163](#)

- [“Canceling key recovery” on page 164](#)

## Setting users for key recovery

You can cancel key recovery at any time before the user completes the key recovery operation. For more information, see [“Canceling key recovery” on page 164](#).

### To set a user for key recovery

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Find the user whose keys you want to recover (see [“Finding users” on page 134](#)).
- 3** Select the user and then select **Users > Selected User > Begin Key Recovery**.
- 4** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, the **Operation Completed Successfully** dialog box appears. This dialog box displays the reference number and authorization code that the user needs to reactivate the profile. Record these activation codes.

- 5** Click **OK**.
- 6** Give these new activation codes to the user. The user needs these codes to complete the reactivation operation. When the user next logs in to the Security Manager client application, they must enter these codes to reactivate their profile.

Activation codes are retained until the user recovers the profile. To view these activation codes, see [“Viewing activation codes and expiry dates” on page 155](#).

## Canceling key recovery

You can cancel key recovery for a user if the user has not reactivated the profile using the new activation codes. Canceling key recovery is done when either of the following events occur:

- the user restores a profile from backup copies
- the user remembers the password
- an Entrust PKI administrator has mistakenly selected the wrong user

### To cancel key recovery

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Find the user whose keys you want to recover (see [“Finding users” on page 134](#)).
- 3** Select the user and then select (**Users > Selected User > Cancel Key Recovery**).
- 4** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

# Recovering user profiles

Entrust PKI administrators with sufficient permissions can complete the key recovery process (see [“Recovering user key pairs” on page 162](#)) by recovering a user profile. Typically, you must recover a user’s profile when it is lost or damaged, or when the user forgets their password.

---

**Note:** This feature is intended for non-human entities, such as Web servers or other hardware devices, or the profiles required to run Security Manager client applications. It is recommended that you do not recover user profiles for users in your organization.

---

Security Manager Administration is a V1 client application. You can recover user profiles with Security Manager Administration for V1 users. For V2 users, you must use a V2 client application (such as Entrust Entelligence Security Provider for Windows) to recover the profile, or you can convert the V2 user to a V1 user (see [“Converting V2 users to V1 users” on page 247](#)).

You can recover user profiles to software or a hardware token.

- [“To recover a user profile to software” on page 165](#)
- [“To recover a user profile to a token” on page 166](#)

## To recover a user profile to software

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Find the user whose profile you want to recover (see [“Finding users” on page 134](#)).
- 3** Select **Users > Selected User > Recover Profile**.
- 4** If Security Manager Administration detects a hardware token, the **Select profile storage location** dialog box appears. Click **No** to store the profile on software. The **Recover profile** dialog box appears.

- 5** To create a desktop profile, click **Create desktop profile** (selected by default). Otherwise, click **Create roaming profile** to create a roaming profile.  
You can only create a roaming profile if a Roaming Server is installed and running.
- 6** In the **Name** field, enter the name for the profile and Entrust support files.  
For example, If you enter `Security Officer`, the profile is named `Security Officer.epf`.
- 7** In the **Location** field, enter a location for the profile and Entrust support files, or click **Browse** to choose a location.
- 8** In the **Password** and **Confirm** fields, enter a password or have the user enter the password for you.  
The password must conform to the password rules. Click **Password Rules** to view the password rules.

---

**Note:** This password is very important. The user uses this password to access the activated profile. Record the password so you can give it to the user.

---

- 9** Give the profile and password to the user.

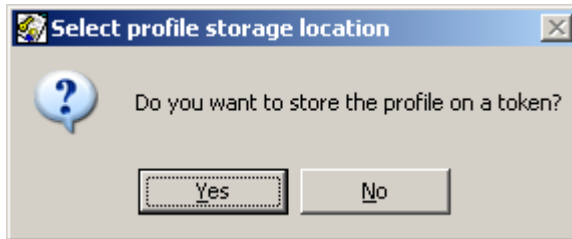
#### To recover a user profile to a token

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Ensure that the token that will contain the profile is in the token reader.

- 3 Find the user whose profile you want to recover (see [“Finding users”](#) on page 134).

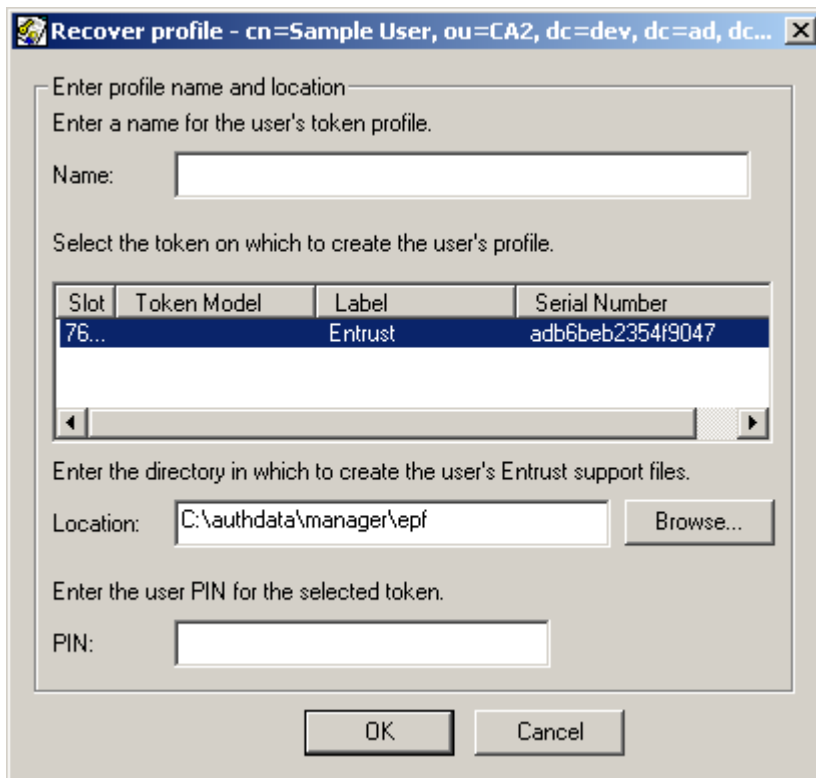
- 4 Select **Users > Selected User > Recover Profile**.

The **Select profile storage location** dialog box appears.



- 5 Click **Yes** to store the profile to a token.

The **Recover Profile** dialog box appears.



- 6 In the **Name** field, enter the name for the profile and Entrust support files. Do not enter a name longer than 75 characters.

- 7 In the token list, select the token where you want to save the token.

- 8** In the **Location** field, enter a location for the Entrust support files, or click **Browse** to choose a location.
- 9** In the **PIN** field, enter the PIN previously assigned to the token.
- 10** Click **OK**.

If there is already a profile on the token, a message appears asking if you want to overwrite the existing profile. Click **No** to cancel the profile recovery, or click **Yes** to overwrite the existing profile.



# Restricting users

You can restrict a user from using Security Manager in a number of ways, each of which is used for different reasons and has different effects. This section describes the methods to restrict user access to Security Manager and explains why you would want to use each method.

This section contains the following topics:

- [“Deactivating users” on page 169](#)
- [“Revoking user certificates” on page 170](#)
- [“Suspending user certificates” on page 171](#)
- [“Archiving and removing users” on page 171](#)

## Deactivating users

When you deactivate a user, that user cannot use key management operations, and the user's information is removed from the directory. This state is usually temporary, so you typically deactivate a user when you know that you want to reactivate the user at a later date (for example, if the user has left on parental leave and plans to return).

Alternatively, you may want to deactivate a user before you impose further restrictions, such as revoking certificates. Once deactivated, you can then archive and remove the user. Archiving a user removes the user's information from the Security Manager database, so that you can then remove the user from the database. (For more information, see [“Archiving and retrieving users” on page 183.](#))

Another common reason for deactivating a non-active user is to reuse the Enterprise license, as the user limit count increases by one when you deactivate a user. Deactivation does not necessarily mean that the user cannot use desktop applications. For example, the user can log in offline with Entrust desktop applications, and other users who have imported that user's encryption certificate can still encrypt for the user. This means that deactivation is not a secure way to stop certificates from being used. If you want to prevent a certificate from being used, you should revoke the user's certificates. However, deactivation does decommission the user so that you can reuse the user's license.

When you deactivate a user without a key history, the user is automatically removed from the database, as there is no information to archive.

---

**Note:** In some situations, Security Manager may deactivate a user as a security precaution. This happens when Security Manager is unable to complete a protocol transaction exchange from a client application three times without success after creating certificates for the user and sending the user to the desktop application. The precaution is designed to operate in network failures, but may also occur because of server overloads, or malformed data messages. The user is deactivated to limit the number of certificates created as these lost certificates are revoked. This type of situation on a large scale can result in large CRLs.

If a user is deactivated for this reason, an audit log records the event. Afterwards, you must reactivate the user.

---

See [“Deactivating and reactivating users” on page 172.](#)

## Revoking user certificates

Another method of restricting a user is to revoke the user's certificates. Typically, you revoke a user's certificates when the user is no longer trusted—for example, if you suspect that an attacker has compromised the user's profile and password. You can also revoke certificates when there is no suspicion of compromise (for example, when a user's distinguished name changes).

You can revoke all or selected certificates, and you can revoke certificates for a user in any state. If you revoke the user's verification certificate, the user is unable to log in to Entrust. If you revoke the user's encryption certificate, Security Manager forces a key update during the next login.

When you revoke the user's certificates, you should reissue the certificate revocation list (CRL) associated with this certificate immediately. You can do this during the revocation operation or afterwards. If you do not reissue the CRL, Security Manager does not update the CRL until it expires, and the certificate revocation does not take effect until after the CRL expires and is updated.

After you revoke the user's certificates, you can put the user into key recovery. This process issues new certificates and enables the user to use Entrust again.

Alternatively, if you revoked the certificates for reasons of cessation of operation—such as a user's termination of employment—you can archive the user's information, and remove the user from the Security Manager database. (For more information, see [“Archiving and retrieving users” on page 183.](#))

See [“Revoking user certificates” on page 174.](#)

---

**Note:** While it is possible to revoke a user's certificate and recover it multiple times, you should avoid this practice. Security Manager automatically returns all key pairs that belong to a particular user certificate as part of a recovery or key update. Multiple key recoveries mean an ever-increasing Entrust profile. A large profile can impact roaming download times and slow down user validation.

---

## Suspending user certificates

If there is any doubt about whether to revoke a user's certificates, you can suspend the user's certificates (put the user on hold). For example, you typically suspend a user's certificates when a user loses a smart card. As in revoking the user's certificates, you should reissue the CRL associated with this certificate immediately.

If the user finds the card later, you can take the user off hold. If the card is not found, you can go ahead and revoke the certificates. See ["Suspending user certificates" on page 181](#).

## Archiving and removing users

Finally, you can remove a user from the Security Manager database by archiving the user's information after deactivation. Typically, you remove a user to save database space in a case in which you do not expect the user to return.

See ["Removing users from the database" on page 191](#).

# Deactivating and reactivating users

Typically, you deactivate a user as a temporary measure. Deactivating users allows you to reactivate the user at a later time, or to remove the user entirely. For an explanation of the different methods of restricting users from access to Entrust, see [“Restricting users” on page 169](#).

This section contains the following topics:

- [“Deactivating users” on page 172](#)
- [“Reactivating users” on page 173](#)

## Deactivating users

Deactivating a user is one method of restricting a user's access to Security Manager (see [“Restricting users” on page 169](#) for other methods). When you deactivate a user, the user can no longer log into the Entrust application online and the user's encryption certificate is no longer available online for access by other users. However, the user's information is kept in the Security Manager database for reactivation at a later date.

---

**Note:** If the user is in the Added state (see [“User states” on page 131](#)), deactivating the user removes all of the user's information, including the information in the Security Manager database.

---

For each Enterprise user you deactivate, the number of available licenses for enabling new users in your organization increases by one. For example, if you have 1000 licenses and they are all used, deactivating a user allows you to add one new user. The total number of licenses (1000) has not changed. For Web licenses, you cannot deactivate a Web user as a way to reuse these licenses. One Web license is permanently used each time a Web certificate is issued.

### To deactivate a user

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Find the user you want to deactivate (see [“Finding users” on page 134](#)).
- 3 Select the user and then select **Users > Selected User > Deactivate**.
- 4 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

## Reactivating users

You can reactivate a previously deactivated user to give the user access to Entrust applications once again.

You can only reactivate a user if there is at least one unused license. You must set the user for key recovery if the user loses the digital ID or forgets the password, or if the user's signing private keys have expired.

### To reactivate a user

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Find the user you want to reactivate (see [“Finding users” on page 134](#)).
- 3** Select the user and then select **Users > Selected User > Reactivate**.
- 4** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

# Revoking user certificates

Deactivating and revoking certificates are related tasks that, together, restrict users from using Security Manager and client applications. For basic information about deactivating and revoking users, see [“Restricting users” on page 169](#).

Most users possess two kinds of user certificates—an encryption certificate and a verification certificate. If you revoke the user’s verification certificate, the user is unable to log in to Entrust. If you revoke the user’s encryption certificate, Security Manager forces a key update during the user’s next login. It is also possible that other users, too, are restricted from encrypting for the user (see [Table 15 on page 174](#)).

---

**Note:** One-key-pair users have a dual-usage certificate, used for both encrypting and verifying data. If you revoke the certificate of a 1-key-pair user, the user cannot log in to Security Manager client applications, and other users may be unable to encrypt data for the user. For more information about 1-key-pair users, see [“1-key pair option” on page 114](#).

---

This section contains the following topics:

- [“Reasons for revoking certificates” on page 174](#)
- [“Login messages received for revoked certificates” on page 175](#)
- [“Revoking certificates” on page 177](#)
- [“Issuing a new CRL after revoking a certificate” on page 180](#)

## Reasons for revoking certificates

When you revoke a user’s certificates you must specify why you revoked the certificates. Table 15 describes the reasons for revoking certificates that you can choose when revoking the certificates.

**Table 15:** Reasons for revoking a user’s certificates

Reason	Definition
Superseded	The certificate is replaced, but there is no suspicion of compromise. Depending on the security policy of your organization, you may wish to revoke non-current certificates after a key update has occurred.
Key Compromise	The private key corresponding to the public key in the certificate is compromised or is suspected to be compromised.

**Table 15:** Reasons for revoking a user's certificates (continued)

Reason	Definition
Affiliation Change	<p>The name or location of the subject of the certificate has changed, but there is no suspicion of compromise.</p> <p>For example, your organization may have a policy to revoke the certificates in a user's profile after a Change DN operation (in which new certificates are generated). In this case, you specify the Affiliation Change revocation reason during the revocation operation.</p> <p>If you revoke the encryption certificate, a key update occurs when the user next logs in. This allows users to automatically complete a Change DN operation the next time they log in.</p>
Cessation of Operation	<p>The certificate is no longer needed for its original purpose, but there is no suspicion of compromise.</p>
On Hold	<p>The user has misplaced a smart card or token, so the certificate is in danger of compromise.</p> <p>Specifying this reason for revocation suspends the certificate.</p>
Unspecified	<p>None of the other revocation reasons apply as to why the certificate was revoked.</p> <p>You can clarify the revocation reason by adding a comment in the <b>Certificate revocation comment</b> field in the <b>Revoke Certificate</b> dialog box.</p>

## Login messages received for revoked certificates

[Table 16 on page 176](#) lists the messages a user receives when logging in to a Security Manager client application after one or more of the user's certificates were revoked.

The results shown in this table depend on the state of certificate revocation list (CRL) caching. If CRL caching is on, the most recent local CRL on disk is not affected until it expires and is updated (after which, the user receives the log in messages shown in this table). When CRL caching is off, the most recent CRL is taken from the directory at login time and the results shown in the table occur immediately.

By default, CRL caching is enabled. To disable CRL caching, open the `entrust.ini` file and set `CrlCachEnabled=0`. For information about the `entrust.ini` file, see the *Security Manager Operations Guide*.

**Table 16:** Login messages received after certificate revocation

Revocation reason	Both certificates	Verification certificate	Encryption certificate
On Hold	Not applicable. (You cannot revoke both certificates for this reason. It is unavailable.)	The user's public or private key information is invalid.	Your profile is automatically updated. If you have made any copies of your profile, do not use those copies any more.
Superseded	The user's public or private key information is invalid.	The user's public or private key information is invalid.	Your profile is automatically updated. If you have made any copies of your profile, do not use those copies any more.
Key Compromise	The user's public or private key information is invalid.	The user's public or private key information is invalid.	Your profile is automatically updated. If you have made any copies of your profile, do not use those copies any more.
Affiliation Change	Your signing certificate is revoked. Please contact your Administrator to recover your profile.	Your signing certificate is revoked. Please contact your Administrator to recover your profile.	Your profile is automatically updated. If you have made any copies of your profile, do not use those copies any more.
Cessation of Operation	The user's public or private key information is invalid.	The user's public or private key information is invalid.	Your profile is automatically updated. If you have made any copies of your profile, do not use those copies any more.
Unspecified	The user's public or private key information is invalid.	The user's public or private key information is invalid.	Your profile is automatically updated. If you have made any copies of your profile, do not use those copies any more.



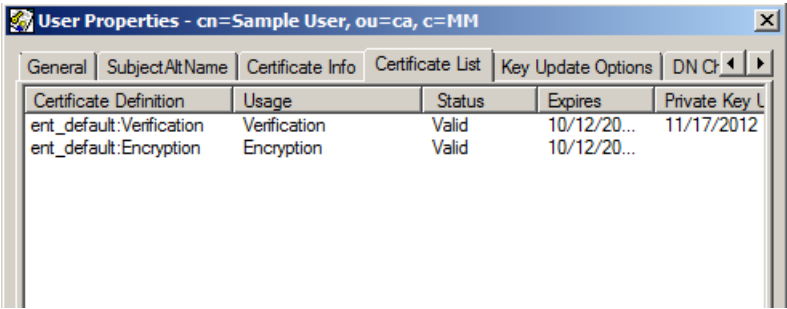
# Revoking certificates

You can revoke one of a user's certificates or all of a user's certificates. Revoke a specific certificate if you want to revoke only one of the user's certificates (for example, if you want to revoke the user's verification certificate but not the encryption certificate).

- ["To revoke a certificate" on page 177](#)
- ["To revoke all user certificates" on page 179](#)

## To revoke a certificate

- 1 Log in to Security Manager Administration (see ["Logging in to Security Manager Administration" on page 46](#)).
- 2 Find the user whose certificate you want to revoke (see ["Finding users" on page 134](#)).
- 3 Select the user and then select **Users > Selected User > Properties**.  
The **User Properties** dialog box appears.



- 4 Click the **Certificate List** tab.

---

**Note:** One-key-pair users use a dual-usage certificate for both encrypting and verifying data. Dual-usage certificates appear in the Usage column of the Certificate List property page as **Encryption/Verification**. For more information about 1-key-pair users, see ["Configuring V1 1-key-pair users" on page 119](#).

---

- 5 Right-click the certificate you want to revoke, select **Revoke this certificate** and then select the reason why you are revoking the certificate (see ["Reasons for revoking certificates" on page 174](#) for a description of each reason).  
The **Revoke Certificate** dialog box appears.

**Revoke Certificate**

Revocation reason: Cessation of operation

Certificate revocation comment:

Enter the date when the certificate was last known to be uncompromised:

☐ Issue the CRL associated with this certificate

OK Cancel Help

- 6** In the **Certificate revocation comment** field, you can enter additional information about why you are revoking the certificate. You can use this field to provide more information about the revocation than just the revocation reason.

You can review the comment later by reviewing the user's properties and viewing the certificate list.

- 7** If you selected **Key Compromise** as the revocation reason, enter the date when the certificate was last known to be uncompromised (when you believe that the certificate was last trustworthy).

Security Manager client applications use the date the key was last known to be uncompromised to verify a signed and timestamped file. If the timestamp precedes the date when the certificate was last known to be uncompromised, the application verifies the signed file.

- 8** Select **Issue the CRL associated with this certificate** if you want to reissue the certificate revocation list (CRL) associated with this certificate immediately.

If you do not select this check box, the CRL is not updated immediately. The certificate revocation does not take effect until after the CRL expires and is updated. The CRL update period is defined by the Security Policy (see ["Configuring the Security Policy" on page 84](#)).

If the CRL policy states that CRLs are always issued after certificate revocation, this option is automatically selected and disabled, ensuring that the CRL is issued. Entrust PKI administrators may not have the correct permissions to issue CRLs.

When you revoke the certificates of an Entrust PKI administrator, it is recommended that you issue a new certificate revocation list (CRL) immediately. See ["Issuing a new CRL after revoking a certificate" on page 180](#).

- 9** If prompted to authorize the operation, authorize the operation. See ["Authorizing sensitive operations" on page 52](#).

If the operation was successful, a success message appears.

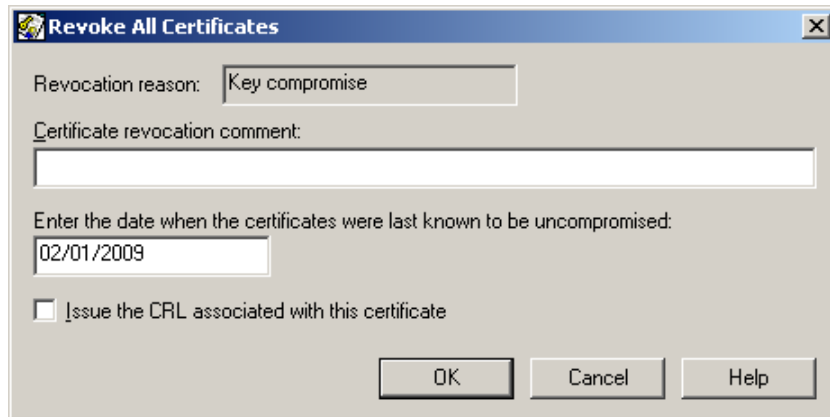
### To revoke all user certificates

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Find the user whose certificates you want to revoke (see [“Finding users” on page 134](#)).
- 3 Select **Users > Revoke Certificates** and then select the reason for revoking the certificates (see [“Reasons for revoking certificates” on page 174](#)).

A dialog box appears asking if you want to continue.

- 4 Click **Yes** to continue.

The **Revoke All Certificates** dialog box appears.



- 5 In the **Certificate revocation comment** field, you can enter additional information about why you are revoking the certificate. You can use this field to provide more information about the revocation than just the revocation reason.

You can review the comment later by reviewing the user's properties and viewing the certificate list.

- 6 If you selected **Key Compromise** as the revocation reason, enter the date when the certificate was last known to be uncompromised (when you believe that the certificate was last trustworthy).

Security Manager client applications use the date the key was last known to be uncompromised to verify a signed and timestamped file. If the timestamp precedes the date when the certificate was last known to be uncompromised, the application verifies the signed file.

- 7 Select **Issue the CRL associated with this certificate** if you want to reissue the certificate revocation list (CRL) associated with this certificate immediately.

If you do not select this check box, the CRL is not updated immediately. The certificate revocation does not take effect until after the CRL expires and is updated. The CRL update period is defined by the Security Policy (see [“Configuring the Security Policy” on page 84](#)).

If the CRL policy states that CRLs are always issued after certificate revocation, this option is automatically selected and disabled, ensuring that the CRL is issued. Entrust PKI administrators may not have the correct permissions to issue CRLs.

When you revoke the certificates of an Entrust PKI administrator, it is recommended that you issue a new certificate revocation list (CRL) immediately. See [“Issuing a new CRL after revoking a certificate” on page 180](#).

- 8** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

## Issuing a new CRL after revoking a certificate

You can issue a new certificate revocation list (CRL) to show the newly-revoked certificates.

### To issue a CRL

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Select **CAs > Issue Updated CRLs**.
- 3** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

# Suspending user certificates

Security Manager lets you permanently revoke certificates when they are no longer trusted ([“Revoking user certificates” on page 174](#)), but you can also suspend user certificates (put them on hold). For example, if a user misplaces a smart card, you can suspend the certificates on that card. If the user finds the card, you can take the certificates off hold. If the user does not find the smart card, you can permanently revoke the certificates. For basic information about methods of restricting users, see [“Restricting users” on page 169](#).

**Note:** The effects of suspending a certificate are significant. A suspended certificate is not trusted in the same way that revoked certificates are not trusted.

Security Manager adds an entry to the CRL for the certificate. The entry indicates that the certificate is suspended, and is not trusted at this time.

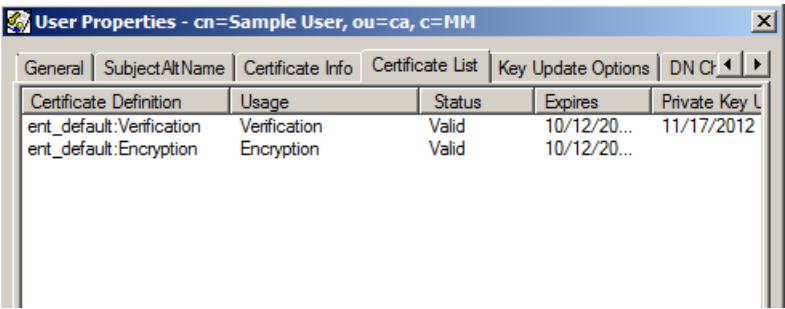
This section contains the following procedures:

- [“To put a user certificate on hold” on page 181](#)
- [“To take a user certificate off hold” on page 182](#)

## To put a user certificate on hold

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Find the user whose certificate you want to revoke (see [“Finding users” on page 134](#)).
- 3 Select the user and then select **Users > Selected User > Properties**.

The **User Properties** dialog box appears.



- 4 Click the **Certificate List** tab.
- 5 Right-click the certificate you want to put on hold, and select **Revoke this certificate > On Hold**.

The **Revoke Certificate** dialog box appears.

- 6** In the **Certificate revocation comment** field, enter a comment that describes why the you are putting the certificate on hold.
- 7** Select **Issue the CRL associated with this certificate** if you want Security Manager to immediately publish a new CRL with the entry that indicates this certificate is suspended.

---

**Note:** Depending on your administrative permissions, you may not be able to select this option. If you cannot set this option, an entry that indicates this certificate is suspended is included in the next CRL issued by default.

---

- 8** Click **OK**.
- 9** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
If the operation was successful, a success message appears.

#### To take a user certificate off hold

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Find the user whose certificate you want to take off hold (see [“Finding users” on page 134](#)).
- 3** Select the user and then select **Users > Selected User > Properties**.  
The **User Properties** dialog box appears.
- 4** Click the **Certificate List** tab.
- 5** Right-click the certificate you want to put on hold, and select **Cancel Hold**.  
The **Cancel Hold** dialog box appears.
- 6** Select **Issue the CRL associated with this certificate** to immediately issue a new CRL in which the entry for the certificate no longer appears, and then click **OK**.
- 7** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
If the operation was successful, a success message appears.

# Archiving and retrieving users

The Security Manager database contains important information about users, such as their decryption private keys. As the number of users who belong to a Certification Authority (CA) increases, so does the size of its database. The size of the Security Manager database affects its performance. For example, the larger the database, the longer it takes to validate and back up the database.

To optimize the performance of the database, you can archive users. When you archive a user, information about that user is removed from the database, and some of that information is written to an archive file. This information includes the critical user data (such as the user's decryption private keys).

Because you can only retrieve some of the information removed from the database, it is recommended that you archive users only when you do not expect them to return to the CA. For example, do not archive a user who is taking parental leave and plans to return to work in a few months.

Because critical information is saved to a file, you can retrieve a user if necessary. For example, if your organization rehires a previously archived user, you can retrieve the critical information about that user from the file. The retrieved information is returned to the Security Manager database. Because new information is added to the retrieved information, the Security Manager database contains a complete history of the information for that user.

This section contains the following topics:

- ["Archive files" on page 183](#)
- ["Archiving users" on page 186](#)
- ["Retrieving archived users" on page 186](#)

## Archive files

When you archive a user, Security Manager removes information about that user from the database and the user's encryption public certificates from the directory. Security Manager does not remove the user entry from the directory.

Security Manager saves some of the information it removes from the database into an archive file. Security Manager stores archive files in the archive folder on the Security Manager server.

The archive file includes the following information:

- the user's decryption private keys and the corresponding encryption certificates
- the user's verification certificates
- any other user certificates for a V2 user

If you retrieve this user, the CA stores each certificate in the database with the same identifier it had in the archive file. If a retrieved V2 user's certificate type no longer exists, the CA sets the certificate type to Enterprise Default.

- a V2 flag to identify a V2 user
- the CA user's verification certificate

If you need to retrieve this user, the CA uses this certificate to verify its own digital signature on the encrypted information.

- the identifier of an encryption key pair

An encryption key pair is made up of a public key used to encrypt information and the corresponding private key used to decrypt that information. Because encryption key pairs are updated at regular intervals, every key pair has an identifier.

If you need to retrieve this user, the CA uses the identifier to determine which private key corresponds to the public key it used to encrypt the information. Only the corresponding private key can decrypt the information.

The archive file does not include user properties, such as group membership, user role, database field values, or `subjectAltName` extensions.

To ensure the security of the information contained in the archive file, Security Manager encrypts the sensitive information and signs all the data before writing it to the archive file. Security Manager encrypts the information using the CA user's encryption public key, and signs the information using the CA user's signing private key.

The name of each archive file is a SHA1 hash of the user's distinguished name (DN). Before hashing the DN, Security Manager converts the DN to lowercase and removes whitespaces. If the DN includes a multi-valued relative distinguished name (RDN), Security Manager will sort the DN per the AVA method before generating the SHA1 hash. For example, if the user's DN is `cn=User+serialNumber=100,o=example,c=US`, then Security Manager will convert the DN to `serialNumber=100+cn=User,o=example,c=US` before applying the SHA1 hash.

The file extension represents the time that the user was archived. The format of the file extension is `YYMMDDhhmmssZ`, where:

- `YYYY` represents the year
- `MM` represents the month
- `DD` represents the day
- `hh` represents the hour
- `mm` represents the minute
- `ss` represents the second
- `Z` indicates that the time is represented using Coordinated Universal Time (UTC).



---

**Attention:** Do not change the names of the archive files or add files of any type to the archive folder, or you cannot retrieve archived users later.

---

## Viewing archive files

You can view the contents of an archive file using a text editor. Because the archived information is encrypted, you cannot read this information. However, the archive file shows the distinguished name (DN) of the archived user in plaintext, which is the only way to determine which archive file belongs to each user. You may want to view the contents of an archive file to locate a particular user's DN. For example, if you want to delete the archive file for a particular user, you can check the DNs in the archive files to locate the one you are looking for.

## Moving archive files

Security Manager creates archive files in the archive folder on the Security Manager server. It is recommended that you periodically move archive files to another location for long-term storage. As the number of archive files in the archive folder increases, more time may be required to complete operations such as backing up the Security Manager data files.

Because the sensitive information in archive files is signed and encrypted, you can move the files using an unsecured method (for example, using Windows Explorer). If you need to retrieve an archived user, you must first move the archive file into the archive folder.

Security Manager also keeps a list of all archived users in the Security Manager database, allowing you to find archived users more quickly. The list contains only the distinguished name (DN) of each user and the name of the user's archive file. When you archive a user, Security Manager creates an archive file for the user and also adds the user to the list of archived users in the database. If you move an archive file, the list of archived users in the database no longer matches the archive files in the archive folder. A Master User must run the `util load-archived-users` command to reload the list of archived users into the database (see the *Security Manager Operations Guide*).

## Deleting archive files

Your organization's security policies may require that you periodically delete archive files. For example, these policies may specify that you delete archive files that are more than one year old.

Because the sensitive information in archive files is signed and encrypted, you can delete the files using an unsecured method (for example, by moving the files to the

Windows Recycle Bin). If you delete an archive file, you cannot retrieve the user later. You must have a user's archive file in the archive folder to retrieve an archived user.

Security Manager also keeps a list of all archived users in the Security Manager database, allowing you to find archived users more quickly. When you archive a user, Security Manager creates an archive file for the user and also adds the user to the list of archived users in the database. If you delete an archive file, the list of archived users in the database no longer matches the archive files in the archive folder. A Master User must run the `util load-archived-users` command to reload the list of archived users into the database (see the *Security Manager Operations Guide*).

## Archiving users

Entrust PKI administrators with sufficient permissions can archive users in the Deactivated or Export states (see ["User states" on page 131](#)). You cannot archive the CA user.

Because you can only retrieve some of the information removed from the database, it is recommended that you archive users only when you do not expect them to return to the CA. For example, do not archive a user who is taking parental leave and plans to return to work in a few months.

Once you archive a user, you cannot perform administrative tasks for that user. Ensure that you have performed any necessary administrative tasks, such as revoking the user's certificates, before you archive a user.

### To archive a user

- 1 Log in to Security Manager Administration (see ["Logging in to Security Manager Administration" on page 46](#)).
- 2 Find the user you want to archive (see ["Finding users" on page 134](#)).
- 3 Select **Users > Selected User > Archive**.
- 4 If prompted to authorize the operation, authorize the operation. See ["Authorizing sensitive operations" on page 52](#).

If the operation was successful, a success message appears.

## Retrieving archived users

You can retrieve an archived user to return the information in the archive file to the Security Manager database. Entrust does not intend that you use this feature frequently. It has been provided only for exceptional circumstances.

You can retrieve an archived user at any time as long as their distinguished name (DN) is not being used by another user. If the DN is being used, the current user represented by that DN must change their DN (see ["Changing distinguished names" on page 193](#)).

After you retrieve an archived user, new information (such as decryption private keys generated when the user logs in for the first time) is appended to the retrieved information. Because new information is appended to retrieved information, the Security Manager database provides a complete history for the user.

---

**Note:** If you simply add the user as a new Entrust PKI user instead of retrieving the archived information, you cannot associate the new information generated for the user with the archived information.

---

Because an archive file does not contain information about the user's properties, Security Manager creates default properties in the same way that it creates properties for a new user. If a retrieved V2 user's certificate type no longer exists, the CA sets the certificate type to Enterprise Default (ent\_default). All V1 users restored from archive get the default certificate type for their category: Enterprise Default (ent\_default) for Enterprise users, or Web Default (web\_default) for Web users.

---

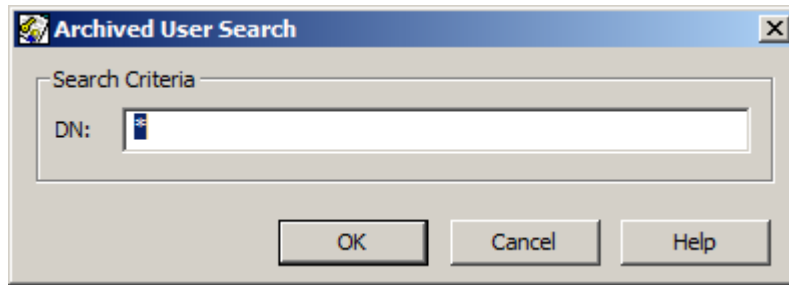
**Note:** If the certificate type assigned to the user has a mandatory database field, the retrieval operation will fail since no value is provided when you retrieve the user. To retrieve the user successfully, the database fields for the certificate type must be optional. For information about setting database fields, see ["Customizing database fields" on page 601](#).

---

### To retrieve archived users

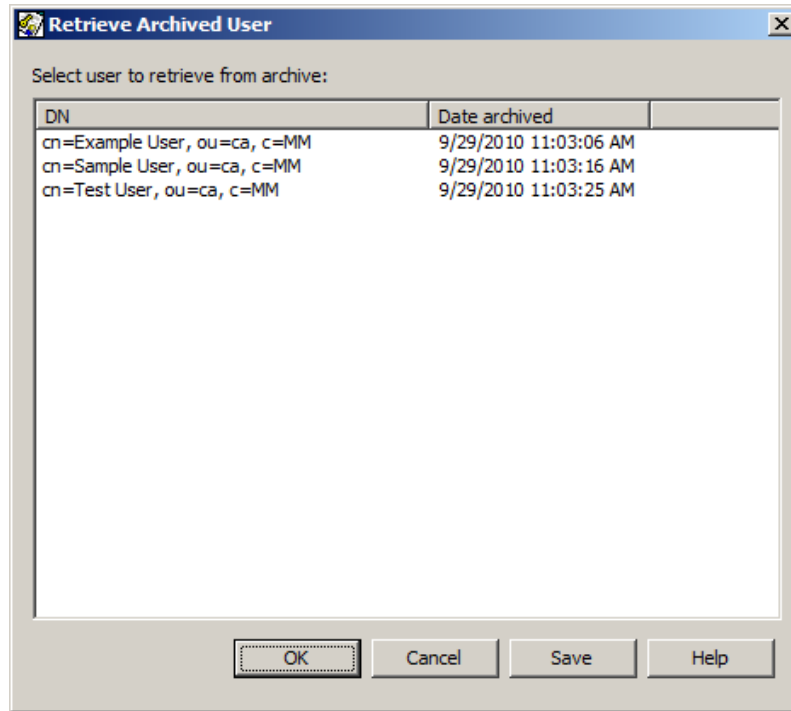
- 1** To retrieve an archived user, there must be a valid archive file in the archive folder on the Security Manager server. If you moved the archive file earlier (see ["Moving archive files" on page 185](#)), you must move it to the archive folder.
- 2** Log in to Security Manager Administration (see ["Logging in to Security Manager Administration" on page 46](#)).
- 3** Ensure that an entry exists in the directory for each user you want to retrieve. For information about creating directory entries, see ["Adding entries to the directory" on page 74](#).
- 4** Select **Users > Archived Users**.

The **Archived User Search** dialog box appears.

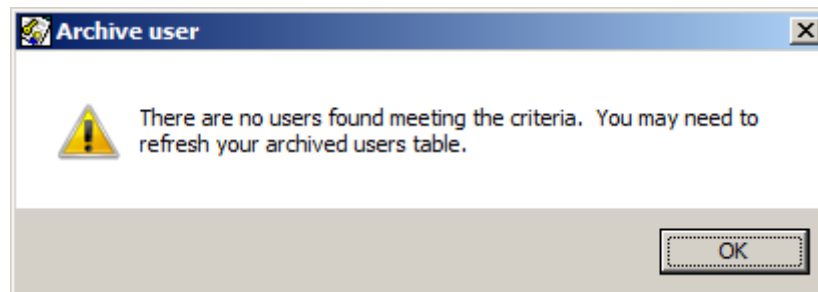


- 5** In the **DN** field, enter the distinguished name (DN) of the user you want to find.  
When you search for entries with directory attribute values that include special characters, the values you enter must match the directory entry.  
Optionally, you can use wildcards. Wildcards let you search for partial attributes. Add an asterisk (\*) with a partial search string to find users that include the search string information. For example, enter \*dr\* to find all entries named Andrew and Drew. Enter dr\* to exclude Andrew and find only entries beginning with the letters dr.
- 6** Click **OK**.

If Security Manager found one or more users that match your search criteria, the **Retrieve Archived User** dialog box appears. This dialog box shows all the archived users who match your search criteria.



If the search found no users that matched your criteria, an **Archive user** dialog box appears, informing you that no users were found that matched the criteria. If you searched for archived users with an empty string or a wildcard character (\*) and no users were found, the **Archive user** dialog box also states that you may need to refresh your archived users table:



If this message appears, one of the following scenarios may apply to your Security Manager:

- No archived users actually exist.

- An archive file for the user may exist in the Security Manager archive folder, but the list of archived users in the Security Manager database has not been updated.

This situation can occur after upgrading or migrating from a previous version of Security Manager, or if you moved archive files into the archive folder without reloading the archived users in the database afterward. A Master User can reload the archived users into the database by running the `util load-archived-users` command (see the *Security Manager Operations Guide*).

- 7** To save the list of archived users to a tab-delimited file, click **Save**.
- 8** Select the user you want to retrieve. To select more than one user, hold the Ctrl key.
- 9** Click **OK**.
- 10** If prompted, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

Security Manager verifies the digital signature on the archive data, decrypts the archive data, and writes the information in the archive file to the Security Manager database.

If the operation was successful, a success message appears. If you retrieved more than one user, a success message appears for each user that Security Manager successfully retrieved.

If Security Manager Administration returns an "Error reading file" error, the error may indicate that the archive file for the user does not exist. This situation can occur if you removed the archive file from the Security Manager archive folder without reloading the archived users in the database afterward. A Master User can reload the archived users into the database by running the `util load-archived-users` command (see the *Security Manager Operations Guide*).

- 11** To reactivate the archived users:
  - If the users were deactivated before being archived, their state will be Deactivated when they are retrieved. To reactivate the users, see [“Deactivating and reactivating users” on page 172](#). If the users' certificates expired, you must recover their keys. To recover their keys, see [“Recovering user key pairs” on page 162](#).
  - If the users were in the Export state before being archived, their state will be Export when they are retrieved. Cancel the export to reactivate the user. See [“Canceling user export operations” on page 213](#) for details.

# Removing users from the database

You can remove users from the Security Manager database, only if the users are in the Added state (see [“User states” on page 131](#)). You cannot remove active users.

To remove an active user, you must deactivate and then archive the user. After archiving a user, you can add the user to Security Manager—putting the user in the Added state—allowing you to remove the user.

When a user is activated, a record of the user’s key history is retained and stored in the Security Manager database, and cannot be deleted. The inability to fully delete a user is a safeguard against accidentally deleting a user’s keys that are needed to open encrypted files. For example, if an important employee leaves your organization, you will probably want access to the former employee’s encrypted files.

When you archive a user (see [“Archiving users” on page 186](#)), information about that user is removed from the database. For example, you may choose to archive users when they leave your organization. The size of the database is minimized because it contains only information about users who belong to the Certification Authority (CA), and are actively working from the CA.

## To remove a user from the Security Manager database

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Find the user you want to remove (see [“Finding users” on page 134](#)).
- 3** You can only remove users in the Added state. Before you can remove active users, you must:
  - a** Deactivate the user (see [“Deactivating users” on page 172](#)).
  - b** Archive the user (see [“Archiving and retrieving users” on page 183](#)).
  - c** Add the user to Security Manager (see [“Adding existing users” on page 152](#)).
- 4** Select the user you want to remove and then select **Users > Selected User > Remove**.  
The **Remove from Entrust** dialog box appears.
- 5** Click **OK** to continue.
- 6** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
If the operation was successful, a success message appears.

# Modifying distinguished names

Entrust PKI administrators with sufficient permissions can modify distinguished names. A distinguished name (DN) is a user's directory name, the standard representation of a directory entry in an LDAP-compliant directory. A DN consists of attributes (such as `cn`, `serialnumber`, `uid`, `o`, and `c`) that define an entry in the directory.

The portion of the DN used to name an entry relative to its parent is called the relative distinguished name (RDN). For example, assume that the following two DNs are defined in the `dc=Company One, dc=com` branch of the directory:

```
cn=Jenny Jessup + uid=0003, dc=Company One, dc=com
cn=John Jones + uid=0005, dc=Company One, dc=com
```

The RDN of the entries makes them unique within the branch. The RDN of the first entry is `cn=Jenny Jessup + uid=0003`. The RDN of the second entry is `cn=John Jones + uid=0005`.

When Security Manager Administration records a DN (for example, when recording results of a bulk operation in a log file), it does not guarantee the order of the RDN attributes. The order of the attributes within a multi-attribute RDN is insignificant. The following example shows two RDN formats for the same DN:

```
cn=Jenny Jessup + uid=0003, dc=Company One, dc=com
uid=0003 + cn=Jenny Jessup, dc=Company One, dc=com
```

Both formats are equivalent according to the LDAP and X.500 standards.

Security Manager Administration records the RDN attributes in a specific order only when defining a DN in a certificate to meet LDAP and X.500 standards, or when adding a new directory entry to avoid directory issues related to inconsistent ordering. In both cases, Security Manager Administration applies distinguished encoding rules (DER).

---

**Note:** Security Manager Administration displays a maximum of 256 characters of a DN, and the Directory Browser displays a maximum of 259 characters of a DN. If any of your users have very long DNs, these DNs might truncate when you view them in Security Manager Administration and the Directory Browser. This limit is a display limit only.

---

This section contains the following topics:

- [“Changing distinguished names” on page 193](#)
- [“Canceling DN changes” on page 198](#)
- [“Assigning new distinguished names” on page 199](#)



## Changing distinguished names

You may need to change a user's distinguished name (DN) for a variety of reasons. For example, you change the user's distinguished name if the user changes their name or moves to another department, or if you need to add or remove an email address from the DN, or correct spelling mistakes.

When you change a DN, you change the DN in the Security Manager database and directory. Changing the DN does not take up another license.

You cannot change a user's DN:

- if the user is in the Deactivated, Export, or Export Hold state (see [“User states” on page 131](#))
- when a key update is pending for a V1 user  
You can change a V2 user's DN if a key update is pending. If you change a V2 user's DN when a key update is pending, Security Manager cancels the key update. V2 client applications will update all user key pairs when it detects the change DN event.  
Policy settings may result in new certificate being generated when a key update is initiated by an administrator. These extra certificates remain if the key update event is canceled and are collected by the V2 client application before it completes the DN change.
- when the **Set key expiry** option is used for the user ([“Configuring user key update options” on page 230](#))

---

**Note:** It is recommended that you do not change the DN of the First Officer. If you change the First Officer's DN, the DN must only contain printable ASCII characters, and cannot contain the following characters:

< > # \ " / | ' ^ ; [ &

You must also change the DN in the `entmgr.ini` file.

---

If you use Microsoft Active Directory as your Security Manager directory, you cannot change a user's DN in Security Manager Administration. You must first change the DN in Active Directory using Microsoft tools, and then assign the new DN to the user (see [“Assigning new distinguished names” on page 199](#)). This limitation does not apply to Microsoft Active Directory Application Mode (ADAM) or Active Directory Lightweight Directory Services (AD LDS).

If you use ADAM or AD LDS as your Security Manager directory, you may need to configure the `entrustra.ini` and `entmgr.ini` files. See the *Security Manager Operations Guide* for more information about these files.

- ADAM and AD LDS contain many read-only attributes that applications cannot copy.

To disable the copy DN option and rename the DN only, open the `entrustra.ini` file and set `ChangedDNFormat=2`.

- ADAM and AD LDS return an error if you try to set naming attributes.

To disable this feature, open the `entmgr.ini` file and set `AddDNAttrsOnRename=0`.

- You cannot retain the old DN.

To disable the **Keep old entry in the Directory** option in the **Change DN** dialog box, open the `entrustra.ini` file and set `DisableRetainOldDN=1`.

- ADAM and AD LDS define `sn` and other attributes as single-valued attributes.

Normally when Security Manager changes the DN, it adds new values to the attribute without deleting or replacing existing values. This causes ADAM and AD LDS to return the error “type or value exists,” which Security Manager treats as success.

You can configure Security Manager so that it checks the attribute values after it attempts to add the new values. If the new value is not in the attribute, Security Manager tries to replace the value. To do so, open the `entmgr.ini` file and set `CheckMissingAttrsOnRename=1`.

---

**Note:** If you modified the user template file to create your own user type, the user's directory entry must match the user type when changing the DN. For example, the directory entry must contain all of the object classes defined in the template. See [“Modifying the user template file and user types” on page 447](#).

---

### To change a distinguished name

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Find the user whose DN you want to change (see [“Finding users” on page 134](#)).
- 3 Select the user and then select **Users > Selected User > Change DN**.

The **Change DN** dialog box appears. An asterisk (\*) appears beside required fields. You must enter information into all fields marked with asterisks.

---

**Note:** To change an attribute in the DN that is not western European (for example, Japanese), see [“Entering international characters in distinguished names” on page 679](#).

---

- 4** In the **First Name** field, enter the user’s first name.
- 5** In the **Last Name** field, enter the user’s last name.
- 6** (Optional.) In the **Serial Number** field, enter the user’s serial number (for example, the employee number).

Depending on your organization, the serial number may not be the employee number. Check with a Security Officer if you are unsure which number to enter.

- 7** (Optional.) In the **Email** field, enter the user’s email address. To enter more than one email address, separate each email address multiple with a space. For more information, see [“Email addresses in Security Manager” on page 132](#).

Due to changing recommendations in the current S/MIME standard, it is recommended that you do not include an email address in a user’s DN. Rather, include the email address in the `subjectAltName` attribute of the user’s X.509 certificate. For more information, see [“Email addresses in Security Manager” on page 132](#).

- 8 In the **Change searchbase to** drop-down list, select the user's new searchbase. For more information about searchbases, see ["Administering searchbases" on page 341](#).
- 9 Choose one of the following options:
  - Select **Keep old entry in the Directory** to keep a copy of the old DN in the directory. (This option is available only if you are using an LDAP v3 directory.)

When you choose to keep the old entry in the directory, a new entry is created in the directory using the new DN and the attributes from the old entry are copied into the new entry. This does not include any non-Entrust managed certificates that are present in the old entry. You must copy these certificates manually. For active users, the Entrust public encryption certificate (which is stored in the `userCertificate` attribute) is added to the user's new directory entry after the user logs in to a Security Manager client application and completed the Change DN operation.
  - Select **Rename existing Directory entry** to replace the existing DN with the new DN. (This option is available only if you are using an LDAP v3 directory.)

When you rename the existing entry, the Change DN operation is implemented with the LDAP `modifyDN` operation. The old RDN of the entry is not deleted. This may cause errors with some directories. If leaving the old RDN in the entry introduces interoperability problems with your directory, you must remove the old RDN from the entry immediately after performing the Change DN operation. You can do this using the Directory Browser (see ["Deleting directory entries" on page 76](#)).

When you select this option, you can choose to keep or delete the old attributes in the entry. Select **Retain old DN values in the entry** to keep the old attributes and the new attributes you assign when you change the user's DN. Deselect **Retain old DN values in the entry** to replace the old attributes with the new ones when you change the user's DN.

---

**Note:** Depending on the directory you are using, you may not be able to select the **Retain old DN values in the entry** option. This occurs when there is no choice—the operation of the directory causes Security Manager to replace the old attributes.

---

- 10 Click **OK** to change the distinguished name.
- 11 If you are changing the DN of a V2 user with a pending key update, a **Change DN** dialog box appears. The dialog box informs you that the user has a key update pending and that changing the DN will cancel the key update. You are asked if you want to continue. Click **Yes** to continue.
- 12 If prompted to authorize the operation, authorize the operation. See ["Authorizing sensitive operations" on page 52](#).

If the operation was successful, a success message appears.

- 13** If you changed the First Officer's DN, open the `entmgr.ini` file and enter the new DN for the `FirstOfficerDN` setting. For example:

```
FirstOfficerDN=cn=First Officer,dc=Company One,dc=com
```

Enter the new DN exactly as it appears in Security Manager Administration.

- 14** Security Manager does not retrieve new `subjectAltName` information from the directory during the DN change. Manually refresh the `subjectAltName` information (see ["Updating subjectAltName component values from the directory" on page 263](#)).

- 15** Depending on your organization's security requirements, you can delete old entries from the directory before or after the user logs in to an Entrust desktop application to complete a change DN operation. See ["Deleting directory entries" on page 76](#).

If you use Microsoft Active Directory, you cannot delete directory entries using the Directory Browser. You must delete directory entries using Microsoft tools.

- 16** Depending on your organization's security requirements, you can revoke the user's old certificates.

If you revoke certificates because of an affiliation change while the user is pending key update, the user can log in and complete the DN change. If you revoke the verification public key certificate for a reason other than an affiliation change, the user cannot log in to complete DN change. In this case you must set the user for key recovery (see ["Recovering user key pairs" on page 162](#)).

You cannot revoke a user's certificates after a DN change if the user is in the Added state since added users do not have certificates to revoke. You cannot revoke a user's certificates after a DN change if the user is in the Import Key Recovery state, because you cannot revoke certificates issued by another CA

If you changed the DN of a user in the Active state (see ["User states" on page 131](#)), the DN in both the Security Manager database and the directory update when the user logs in to a Security Manager client application. Until the user logs in, a DN change is pending for the user.

If you change the DN of a user in the Added or Imported state, the user does not need to complete the DN change by logging in to a client application, since the CA has not yet issued certificates to the user. Changing a user's DN removes the old DN from the Security Manager database. All other user properties, including activation codes, are unaffected by the DN change.

## Troubleshooting DN changes

In some cases when you change the DN of a person using a deployed Entrust client or desktop application, that user does not get prompted as expected with a message that their profile information was updated. This can occur if that end user has not

successfully logged in to an application that uses Security Manager at least once before the DN change was made. To fix this problem:

- 1** Cancel the DN change as described in [“Canceling DN changes” on page 198](#).
- 2** Have the end user log in.
- 3** Change the end user's DN as described in [“Changing distinguished names” on page 193](#).
- 4** Have the end user log out and log in.

At login, the user should be prompted with a message stating that profile information is updated.

If you want to change a user's DN a second time in 24 hours, the user must complete one of the following to complete the DN change:

- Log in to the desktop application to complete the first DN change, and then log out. Log in again (to clear the CertificatePublicationPending flag), and then log out. After you make the second DN change, the user can then log in to the desktop application to complete the second DN change.
- Wait 24 hours after the first DN change has been completed before logging in to complete the second DN change.

## Canceling DN changes

You can cancel a DN change for any user with a pending DN change. You must cancel a DN change before the user logs in to a Security Manager client application. When a user who has a pending DN change logs in, the DN updates.

When you cancel a DN change, the new entry becomes a Non-Entrust PKI user. However, the original user DN entry is now Active and can use Security Manager as usual.

You cannot cancel a DN change for an added or imported user. If you need to change the DN again while the user is still in the Added or Import Key Recovery state, change the user's DN again (see [“Changing distinguished names” on page 193](#)). You can change an added or imported user's DN an unlimited number of times.

---

**Note:** If you are using Security Manager with Microsoft Active Directory, ensure that you rename the DN in the Active Directory using Microsoft tools, if necessary, before canceling the DN change in Security Manager Administration.

---

### To cancel a DN change

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).

- 2 Find the user whose DN change you want to cancel (see [“Finding users” on page 134](#)).
- 3 Select the user and then select **Users > Selected User > Cancel DN Change**.
- 4 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

## Assigning new distinguished names

The distinguished name (DN) is the link between the Security Manager database and the directory. The DN must be the same in both places. Consequently, if a DN changes in the directory, you must update it in the Security Manager database using Security Manager Administration.

You can assign a new DN to added and imported users as well as active users. When you assign a new DN, you change the DN in the Security Manager database only.

You cannot assign a new DN:

- if the user is in the Deactivated, Export, or Export Hold state (see [“User states” on page 131](#))
- when a key update is pending for a V1 user

You can assign a new DN to a V2 user if a key update is pending. If you assign a new DN to V2 user when a key update is pending, Security Manager cancels the key update. V2 client applications will update all user key pairs when it detects the assign DN event.

Policy settings may result in new certificate being generated when a key update is initiated by an administrator. These extra certificates remain if the key update event is canceled and are collected by the V2 client application before it assigns the new DN.

- when the **Set key expiry** option is used for the user ([“Configuring user key update options” on page 230](#))

---

**Note:** It is recommended that you do not assign a new DN to the First Officer. If you assign a new DN to the First Officer, the DN must only contain printable ASCII characters, and cannot contain the following characters:

< > # \ " / | ' ^ ; [ &

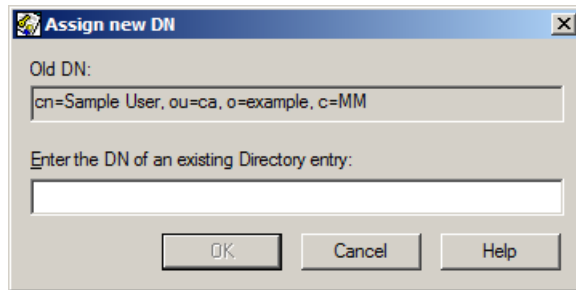
You must also change the DN in the `entmgr.ini` file.

---

### To assign a new DN

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).

- 2 Find the user for whom you want to assign a new DN (see [“Finding users” on page 134](#)).
- 3 Select the user and then select **Users > Selected User > Assign new DN**.  
The **Assign new DN** dialog box appears. The current DN appears in the **Old DN** field.



- 4 In the **Enter the DN of an existing Directory entry** field, enter the new DN
- 5 Click **OK**.  
If the DN you enter cannot be assigned because there is no corresponding directory entry, a message appears.
- 6 If you are assigning a new DN to a V2 user with a pending key update, another **Assign new DN** dialog box appears. The dialog box informs you that the user has a key update pending and that assigning the new DN will cancel the key update. You are asked if you want to continue. Click **Yes** to continue.
- 7 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
If the operation was successful, a success message appears.
- 8 If you assigned a new DN to the First Officer, open the `entmgr.ini` file and enter the new DN for the `FirstOfficerDN` setting. For example:  
`FirstOfficerDN=cn=First Officer,dc=Company One,dc=com`  
Enter the new DN exactly as it appears in Security Manager Administration.



# Restoring user certificates to the directory

You can restore a user's certificates to the directory if the certificates are accidentally deleted or corrupted.

You should back up your directory on a regular basis using your directory backup tools. The procedure described in this section only restores certificate information. If any other type of directory information becomes corrupt or is lost, such as object classes, directory entries, or directory attributes, you can only retrieve this information from a backup generated by your directory backup tools. For more information about backing up your directory, see the *Security Manager Operations Guide*.

## To restore a user certificates to the directory

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Find the user whose certificates you want to restore (see [“Finding users” on page 134](#)).
- 3** Select the user then select **Users > Selected User > Restore Entrust Info to the Directory**.
- 4** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

# Moving users to a new Certification Authority

A Certification Authority (CA) enforces security policies for the users that belong to it. Security policies are reflected in such Entrust settings as the administration policies, password rules, and login policies. For more information about these Entrust settings, see [“Administering user policies” on page 391](#).

Depending on the business requirements or your organization, you may use more than one CA, and you may customize the security policies for each CA. For example, your organization may use a different CA to manage each branch office, and each CA may enforce a different set of password rules (for example, passwords expire every six months for one CA and passwords expire every three months for another CA).

Because a CA enforces security policies for the users that belong to it, and because you can customize security policies for each CA, it is important that each user belongs to the CA with the security policies that are appropriate for that user. This means that when a user's position, function, or location within your organization changes, you may need to move the user to another CA.

For example, suppose that one CA manages the Boston branch office and a different CA manages the Chicago branch office. If a user transfers from the Boston branch office to the Chicago branch office, you should move the user from the CA for the Boston office to the CA for the Chicago office. This move ensures that the security policies of the CA for the Chicago office are enforced for the user. You no longer want the CA for the Boston office enforcing its security policies for the user.

It is recommended that you move a user only between cross-certified CAs (CAs that have a network or hierarchical trust relationship). When you move a user, sensitive information transfers from the old CA to the new CA. Although you can move users between CAs that are not cross-certified, Entrust intends this to occur only when system limitations prevent CAs from cross-certifying (for example, you cannot connect the directories used by the CAs).

This section contains the following topics:

- [“Moving users versus creating users” on page 203](#)
- [“Summary of steps for moving users” on page 204](#)
- [“Establishing trust between the Certification Authorities” on page 206](#)
- [“Exporting users” on page 210](#)
- [“Viewing import files” on page 212](#)
- [“Canceling user export operations” on page 213](#)
- [“Handling user information” on page 213](#)
- [“Importing users” on page 215](#)
- [“Logging in to the new CA” on page 216](#)
- [“Completing user export operations” on page 217](#)

## Moving users versus creating users

At first, it may seem that the easiest way to move a user is to simply deactivate the user in the old CA and add the user in the new CA. The problem with this method is that the user in the new CA has no convenient and safe way to access the encrypted data in the old CA. (Encrypted data in the old CA is referred to as previously encrypted data.)

To access previously encrypted data, users require their decryption private keys from the old CA. Without these keys, the only way for a user to access previously encrypted data is to transfer it to the new CA in an unencrypted state. However, the security risk of leaving sensitive data unencrypted is unacceptable.

Users added to a CA as new users can keep their private decryption keys from another CA only by maintaining two Entrust profiles:

- the profile used in the old CA contains the decryption private keys necessary to access encrypted data in the old CA
- the profile used in the new CA contains the decryption private keys necessary to access encrypted data in the new CA

Maintaining two profiles, however, is inconvenient. For example, suppose that your organization encrypts email messages sent internally. To read messages that were encrypted in the old CA, the user would first have to log in to the CA using the profile for that CA.

### Moving a user's decryption private keys

To provide a safe and convenient way for a user to maintain access to previously encrypted data, Security Manager lets you move an Enterprise user from one CA to another CA (it lets you move a user's decryption private keys to another CA in a safe manner). After moving to the new CA, a user has a single Entrust profile that is used to access data that was encrypted in both the old CA and in the new CA, and that is recoverable if it becomes corrupt or lost.

---

**Note:** You can only move Enterprise users from one CA to another CA. You cannot move Web users from one CA to another CA.

---

## Moving a user's certificates

Security Manager also moves the user's valid encryption and verification certificates to the new CA to provide a record of the kind of certificates that were issued to the user by the old CA.

The new CA may also use the most recent verification certificate to verify the user's digital signature when the user logs in for the first time. For more information about how the new CA uses the most recent verification certificate, see ["Logging in to the new CA" on page 216](#).

---

**Note:** You cannot export roaming users. Attempting to export roaming users results in errors. To export roaming users, you must first change them into desktop users.

---

## Summary of steps for moving users

The steps to move a user involve at least one Entrust PKI administrator at the old CA and at least one Entrust PKI administrator at the new CA. The number of Entrust PKI administrators required at each CA depends on how many Entrust PKI administrators are required to authorize sensitive operations. For more information, see ["Authorizing sensitive operations" on page 52](#).

The following procedure summarizes the steps for moving a user and identifies where you can obtain more information about each step. The tasks in each step and the order of the steps differ slightly depending on which Entrust application the user uses at the new CA.

### To move a user from an old CA to a new CA

- 1 The Entrust PKI administrators at both the old and new CAs establish trust between their CAs.

For information about establishing trust for moving a user, see ["Establishing trust between the Certification Authorities" on page 206](#).

---

**Note:** If you have established trust between CAs by exchanging the CA users' public keys offline, make sure that neither CA has updated its CA keys since trust was established. If it has, you must re-establish trust. (See ["Establishing trust between the Certification Authorities" on page 206](#).)

---

- 2 The Entrust PKI administrator at the old CA collects the information that is required by the new CA, and provides the information to the Entrust PKI administrator at the new CA. This process is called exporting a user.

For information about exporting a user, see ["Exporting users" on page 210](#).

- 3** For Security Provider users moving between cross-certified CAs running Security Manager, the move is handled automatically, as long as the new CA has configuration data present in the user's registry and there is directory connectivity between the directories of the old and new CAs.

For Security Provider users moving between CAs that do not trust each other, or that do not have directory connectivity, the user must have their keys recovered at the new CA. In this case, the configuration settings for the new CA must also be in the registry. For more information, see your Security Provider documentation.

Users of other Security Manager client applications need an updated `entrust.ini` file that points to the new CA before logging in for the first time. The Entrust PKI administrator at the old CA provides the user with an edited `entrust.ini` file. For information about editing the `entrust.ini` file, see the *Security Manager Operations Guide*.

- 4** The Entrust PKI administrator at the new CA writes the information received from the Entrust PKI administrator at the old CA to the new CA.

This process is called importing a user. For information about importing a user, see ["Importing users" on page 215](#).

- 5** For Security Provider users moving between cross-certified CAs running Security Manager, the Entrust PKI administrator at the new CA informs the Entrust PKI administrator at the old CA that the user has been imported.

The Entrust PKI administrator at the old CA completes the user export. For information about completing the export, see ["Completing user export operations" on page 217](#).

In the case of other Security Manager client application users, the user logs in to the new CA. (Some users are also required to recover their profiles before logging in to the new CA.) For information about logging the user in to the new CA, see ["Logging in to the new CA" on page 216](#).

- 6** For Security Provider users moving between cross-certified CAs running Security Manager, the user logs in to the new CA, and the move occurs automatically.

For other Security Manager client application users, the Entrust PKI administrator at the new CA informs the Entrust PKI administrator at the old CA that the user has been imported when the user successfully logs in to the new CA. The Entrust PKI administrator at the old CA completes the user export. For information about completing the export, see ["Completing user export operations" on page 217](#).

When you complete these procedures, a user is moved from the old CA to the new CA.

## Establishing trust between the Certification Authorities

You move users between CAs that have a trust relationship. CAs that have a trust relationship can access the keys necessary to protect sensitive information (the user's decryption private keys and public certificates) during the move. The keys used to protect sensitive information during a move are the CA user keys.

---

**Note:** Do not confuse the CA user keys with the CA signing private key, which can only sign certificates, CRLs, and ARLs.

---

To protect sensitive information during a move, the old CA

- encrypts the sensitive information using the new CA user's encryption public key
- digitally signs the sensitive information using its own CA user's signing private key

The old CA always has access to its own CA user's signing private key. The new CA user's encryption public key, however, is contained in the new CA user's encryption certificate, which is available to the old CA only if a trust relationship exists between the CAs. For the purposes of moving a user, you can establish trust between the CAs

- by cross-certifying the CAs online (see ["Cross-certifying online" on page 474](#)) or by creating a strict hierarchy of CAs online (see ["Creating subordinate CA certificates" on page 510](#)).

The old CA can access the new CA user's encryption certificate from the new CA's directory.

- by exchanging the CA user's public keys offline.

It is strongly recommended that CAs exchange their CA user's public keys only if system issues prevent the CAs from establishing trust online. For example, if the CAs are using different directories, the CAs may not be able to establish the directory connectivity required for online cross-certification. When trust is established through CA public-key exchange, the CAs' directories are not connected. This means that a CA cannot check whether the other CA user's certificates have expired, or whether they are updated or revoked.

### Exchanging the CA public keys

A CA's public keys include its encryption public key and its verification public key. When two CAs exchange their public keys, each CA has the other CA's public keys. Exchanging CA public keys establishes trust between the CAs in an offline manner.

Entrust PKI administrators with sufficient permissions can exchange CA public keys. To exchange CA public keys, you export your CA's public keys to a file that the other administrator imports into the other CA. The other administrator then exports the other CA's public keys to a file and you import them into your CA.

---

**Note:** If you update your CA keys after you export and import them at another CA, you must repeat the export and import process. Similarly, if you have imported keys from another CA, you must import them again if they are updated at the other CA.

---

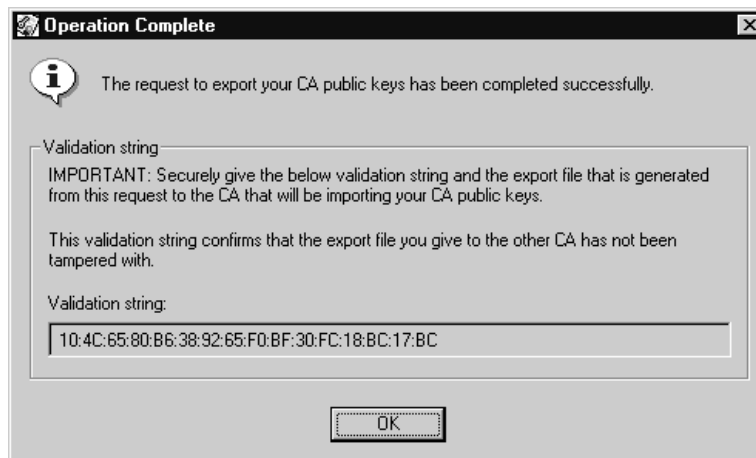
Procedures in this section:

- [“To export the CA public keys from your CA” on page 207](#)
- [“To import the CA public keys from another CA” on page 208](#)
- [“To delete an imported CA” on page 209](#)

### To export the CA public keys from your CA

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Select **Imported CAs > Export CA Public Key**.  
The **Save As** dialog box appears.
- 3 Enter a name for the file and choose a location where you want to save the file. Save the file with a **.cert** extension. By default, the file includes the **.cert** extension.

Security Manager writes the CA public keys to a file and generates a validation string. A dialog box similar to the following appears.



- 4 Record the validation string and click **OK**.
- 5 Give the file and validation string to the administrator of the other CA. It is strongly recommended that you send the validation string and file separately to decrease the chance of tampering or stolen data. For extra security, use different methods to transfer the information. For example, use email to send the file to

the other administrator and use the telephone to inform the other administrator of the validation string.

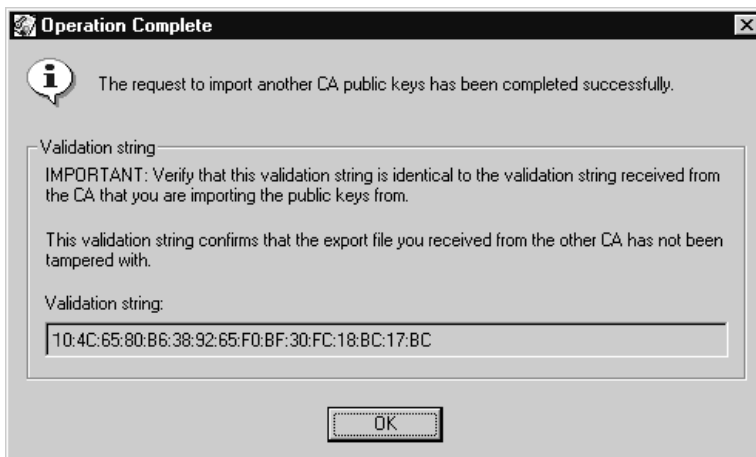
### To import the CA public keys from another CA

- 1 Obtain the CA public key file and validation string from the administrator of the other CA.
- 2 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 3 Select **Imported CAs** > **Import CA Public Key**.

The **Open** dialog box appears.

- 4 Open the file containing the other CA's public keys.

A dialog box similar to the following appears.



- 5 Compare the validation string in the dialog box to the one you received from the Entrust PKI administrator at the other CA.

If the validation strings do not match, delete the file and ask the PKI administrator at the new CA to export the CA user's public keys again. The file may be corrupt or may have been tampered with.

- 6 If the validation strings match, click **OK**.

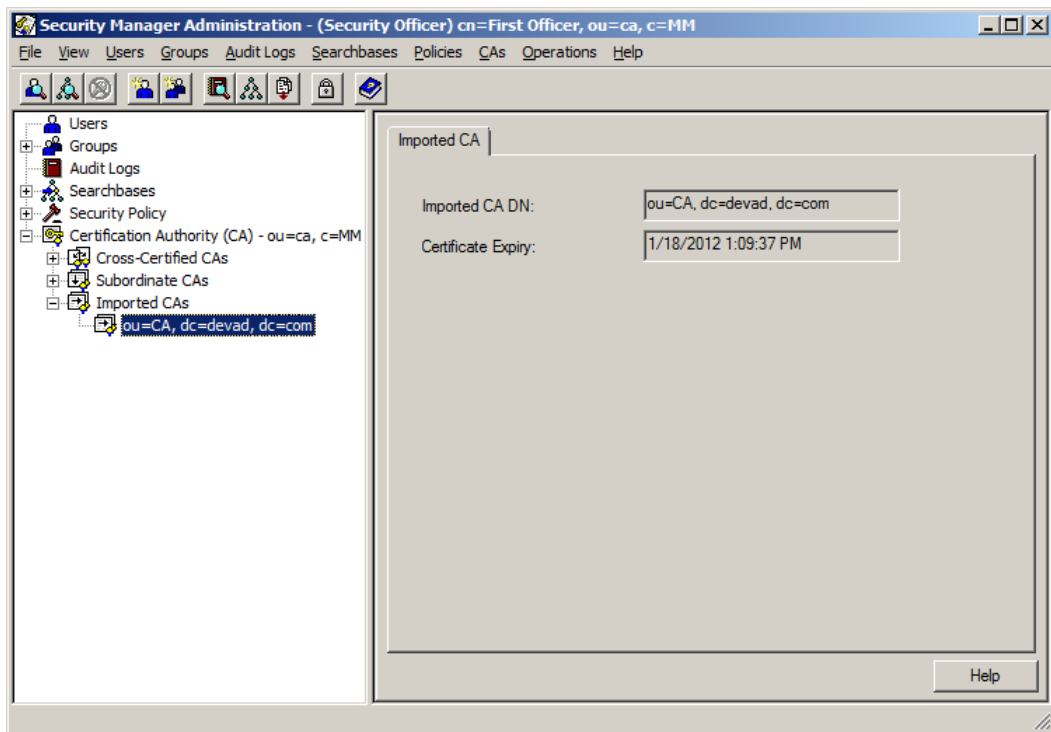
Security Manager imports the contents of the file. If you have previously imported public keys from this CA, the older keys are overwritten.

A dialog box appears when the keys are imported from the file.

- 7 Click **OK**.

The Security Manager Administration window shows the CA under **Imported CAs** in the tree view.





### To delete an imported CA

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 In the tree view, expand **Certification Authority > Imported CAs**.  
The list of imported CAs appears.
- 3 Right-click the CA you want to delete and select **Delete**.

## Exporting users

After establishing trust between the old CA and the new CA (see [“Establishing trust between the Certification Authorities” on page 206](#)), Entrust PKI administrators with sufficient permissions can export users from the old CA. You can only export users in the Active or Deactivated states.

Exporting a user produces an import file that contains the user's decryption private keys and verification certificates from the old CA. The information in the import file is encrypted and signed using the old CA's public key pairs. The administrator at the new CA needs this file to import the user in the new CA.

Because the information in the import file is encrypted and signed, it also includes:

- the old CA's verification certificate  
When the new CA processes the import file, it uses this certificate to verify the old CA's digital signature on the encrypted information.
- the identifier of an encryption key pair  
When the new CA processes the import file, it uses the identifier to determine which private key corresponds to the public key used by the old CA to encrypt the information. Only the corresponding private key can decrypt the information.

---

**Note:** An encryption key pair is made up of a public key used to encrypt information and the corresponding private key used to decrypt that information. Because encryption key pairs are updated at regular intervals, every key pair has an identifier.

---

The import file does not include user properties from the old CA, such as group membership, user role, database field values, or `subjectAltName` extensions. The import file includes group membership and role for the new CA, which you specify when you export the user from the old CA.

After you export the user, the user is placed in the Export Hold state. The number of available Entrust licenses remains the same until the new CA completes the export procedure. If necessary, you can cancel the export operation while the user is in the Export Hold or Export state (see [“Canceling user export operations” on page 213](#)).

When you export an activated user, Security Manager does not remove the user's certificates from the directory. These users can continue to work in the old CA, meaning that their work is not interrupted by the move. However, to ensure that the exported information is complete (no more decryption private keys or verification certificates are created after the user is exported), the old CA does not perform any more key updates. Note that deactivated users cannot work on the CA before or after you export them. Security Manager removes their certificates from the directory when they are deactivated.

A Security Officer in the Export Hold state cannot log in to Security Manager Administration as an active user and perform operations.

Because the old CA does not perform any more key updates for an exported user, it is recommended that the Entrust PKI administrator for the new CA import the user as promptly as possible. If the user's keys expire in the old CA before you import the user into the new CA, the user is unable to log in to any CA.

---

**Note:** You cannot export roaming users. Attempting to exporting roaming users results in errors. To export roaming users, you must first change them into desktop users.

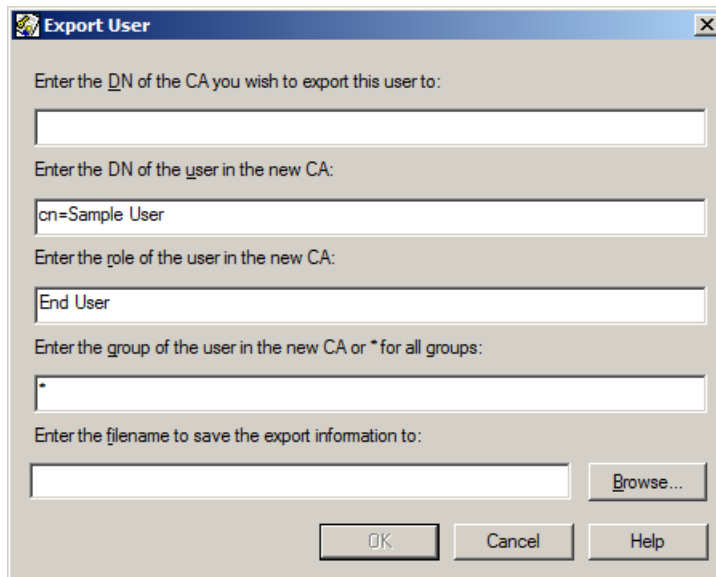
---

To export more than one user at a time, see ["Performing bulk operations" on page 275](#) and ["Bulk commands reference" on page 651](#).

### To export a user

- 1 Log in to Security Manager Administration (see ["Logging in to Security Manager Administration" on page 46](#)).
- 2 Find the user that you want to export (see ["Finding users" on page 134](#)).
- 3 Select **Users > Selected User > Export**.

The **Export User** dialog box appears.

The image shows a Windows-style dialog box titled "Export User". It contains five text input fields and a "Browse..." button. The first field is empty. The second field contains "cn=Sample User". The third field contains "End User". The fourth field contains an asterisk "\*". The fifth field is empty. At the bottom are "OK", "Cancel", and "Help" buttons.

Export User

Enter the DN of the CA you wish to export this user to:

Enter the DN of the user in the new CA:

Enter the role of the user in the new CA:

Enter the group of the user in the new CA or \* for all groups:

Enter the filename to save the export information to:

Browse...

OK Cancel Help

- 4 In the **Enter the DN of the CA you wish to export this user to** field, enter the distinguished name (DN) of the new CA.

- 5** In the **Enter the DN of the user in the new CA** field, enter the DN that the user will have in the new CA.

By default, this field contains the user's current common name (CN). When you click on this field, Security Manager Administration appends the DN of the new CA to the user's current CN. If you do not know the user's new DN, accept the default value.
- 6** In the **Enter the role of the user in the new CA** field, enter the role the user will have in the new CA. By default, the role is End User.
- 7** In the **Enter the group of the user in the new CA or \* for all groups** field, enter the name of the group to which the user will belong in the new CA, or enter an asterisk (\*) to assign the user to all groups.
- 8** In the **Enter the filename to save the export information to** field, enter the full pathname of the file with an `.entra` extension, or click **Browse** to select a location. You must save the file with an `.entra` extension (such as `user_import.entra`).
- 9** Click **OK**.
- 10** If prompted to authorize the operation, authorize the operation. See ["Authorizing sensitive operations" on page 52](#).

If the operation was successful, a success message appears. Security Manager creates the file and writes the user's information into the file in bulk syntax.

To continue exporting the user, give the import file created by Security Manager Administration to the Entrust PKI administrator at the new CA. Because the information in the file is signed and encrypted, you can transfer the file using an unsecured method, such as email.

## Viewing import files

You can view the contents of an import file to determine which users are included in the file. To view the contents of an import file, open the file in a text editor.

When you open an import file, you can see the users' distinguished names (DNs) and the signed and encrypted information (although the information is unreadable). The information includes:

- each user's DN, group information, and role
- each user's key history

The exported key history is an encoded and encrypted string of information much like the private keys in a user profile file.

The import file does not include user properties from the old CA, such as group membership, user role, database field values, or `subjectAltName` extensions. The import file includes group membership and role for the new CA, which you specified when you exported the user from the old CA (see ["Exporting users" on page 210](#)).

Figure 5 on page 213 shows the information for one user in an import file.

**Figure 5:** Example of an import file



```
File Edit Format Help
user x "cn=Nicolò Pelù,dc=Company Three,dc=com"
user_setproperty x group +
user_setproperty x role "End user"
user_import x {
MIITSQIBAZCCE0IGCSqGSIB3DQEHAgCCEZMwghMvAgEBMQkwBwYFKw4DAHOWgg7CBGkqh
kiG9w0BBWgGgg6zBIIOrZCCDqswgg2jBgkqhkiG9w0BBWoggg2UMIINKAIBATGB+ZCB+A
IBATBTMESxEZARBgoJk1aJk/ISZAEZFgNjb20xFDASBgqJk1aJk/ISZAEZFgRHy21IMRA
wDgYDVQQLewduZXR3b3JrMQwwCgYDVQQLewNjYTECBBD4tno4wGwYJKoZIhVZ9B0IKMA4E
CAAAAAAAAAAAAgIAgASBgBZnd+7gOL8ruyh6jDZzAES2ekeBe9vq7Dc6mZQswWPDzqXA
t3sfbNfy1frpeZhn09ekB1CY1/dkN4M4ebQax0wPhdV7iJ8BWUH1+Bz4nnu7n5tXPxPQ7
AptYpq1nb07U2kC1sNnB86rqsJ7x0vialG8KOCxwG/pcvk72/HBfa/MIIM1wyJKoZIhvc
NAQCBMBSGCSqGSIB2fQdCCjAQBAGAAAAAAAAAAAAICAICAggxfMIIMwZCCA0AGC1qGSIB3
DQEJfGgggMwBIIIDLCCAYgwgGKR0AMCAQICBD45P9wwDQYJKoZIhvcNAQEFBQAwmjETM
BEGCgmsJomT81xkArkWA2NvbTEbMBkGCgmsJomT81xkArkWC0NvbXBhbmkgt25lMB4XDT
AZMDEZMDE0NTQyMFoXDTA2MDEZMDE1MjQyMFowRZETMBEGCgmsJomT81xkArkWA2NvbTE
bMBkGCgmsJomT81xkArkWC0NvbXBhbmkgt25lMRMwEQYDVQQDEwppBg1jZSBHcmF5MIGF
MA0GCSqGSIB3DQEBAQUAA4GNADCB1QKBgQC+E1lCg8eTYNhsen+mg7Ac02pBeG/exvLO
```

## Canceling user export operations

If you decide that you do not want to export a user after writing the user information to the import file, you can cancel the export. For example, you can cancel the export when you exported the wrong user or you no longer need to export the user (the user is no longer moving to a new CA). You can cancel the export operation, which returns the user to the state the user was in before the export procedure. For example, if the user was active before the export operation, the user becomes active again when you cancel the export operation.

You can only cancel export operations for users in the Export or Export Hold states.

### To cancel an export operation

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Find the user whose properties you want to change (see [“Finding users” on page 134](#)).
- 3 Select the user and then select **Users > Selected User > Cancel Export**.
- 4 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

## Handling user information

For Security Provider users moving between cross-certified CAs running Security Manager, the move is handled automatically. For Entrust Entelligence Security

Provider users moving between CAs that do not trust each other, you must recover the users' keys at the new CA.

Security Provider does not use the `entrust.ini` file. Instead, the administrator at the new CA must edit the registry file to ensure the move. For more information, see your Security Provider documentation.

Users of other Security Manager client applications require an updated `entrust.ini` file that points to the new CA, as these applications use the `entrust.ini` file to locate the new CA.

In some cases, you can also set the `MovingDomain` flag in the `entrust.ini` file so that users do not need to recover their profile before logging in to the new CA the first time. You should give users the updated `entrust.ini` file before they log in for the first time at the new CA.

If the two CAs are cross-certified (the destination CA can verify the signature on a CMP message signed with a key from the source CA), you can enable automatic key recovery by adding `MovingDomain=1` to the user's `entrust.ini` file and then having the user log in to the Entrust profile.

If the two CAs are not cross-certified, the user must perform manual key recovery by entering the activation codes (generated when the user was imported). In the case of manual key recovery, you must not include the `MovingDomain` setting in the `entrust.ini`. If it is included, you see an error in the `manager.log` indicating that the user is not in the correct state.

---

**Note:** After the user has successfully logged in to the new CA, you must set the `MovingDomain` flag in the user's `entrust.ini` file back to 0.

---

### To edit the `entrust.ini` file

- 1 Open the `entrust.ini` file in a text editor.
- 2 Locate the `[Entrust Settings]` section.
- 3 Change the IP address (or DNS name) and port for the `Authority=` entry to the settings for the new CA:

`Authority=<IP address>+<port>`

where `<IP address>` represents the address or DNS name, and `<port>` represents the port number used by Security Manager applications to connect to Security Manager. For example:

`Authority=1.2.3.4+389`

or

`Authority=NameOfMachine+389`

- 4 Change the IP address and port for the `Server=` entry to the settings for the new CA:

```
Server=<IP address>+<port>
```

where `<IP address>` represents the IP address and `<port>` represents the port number used by Security Manager applications to connect to an LDAP-compliant directory (or LDAP servers fronting a directory).

- 5 If the old and new CAs are cross-certified, enter the following on a new line below the `Server=` entry:

```
MovingDomain=1
```

- 6 Save and close the `entrust.ini` file.

## Importing users

Once the Entrust PKI administrator at the old CA exports a user and gives you the import file, you can import the user. You can import a user if you are a Security Officer for the new CA or if you have the **Import new user** permission (see [“Permissions reference” on page 371](#)).

After receiving an import file from the administrator at the old CA, you can import the users to your CA. You can only import users if your role contains the **Import new user** and **Process bulk files** permissions (see [“Permissions reference” on page 371](#)).

Before you process the import file, your Security Manager directory must include entries for each user you are importing. You can create a directory entry using the Directory Browser, a separate bulk command file, or a third-party tool.

### To import users

- 1 Create entries in your Security Manager directory for each user you are importing.

For information about how to create directory entries using bulk scripts, see [“Creating customized directory entries in bulk” on page 296](#). To create the directory entries properly, remove all lines in the sample script that appear after the `user_createdirentry` command. Removing these lines prevents activation of the user in the new CA. Users are added to the new CA along with their key histories when you process the import file.

- 2 Open the import file that you received from the administrator at the other CA using a text editor. See [“Viewing import files” on page 212](#).

---

**Attention:** Do not change a single character of the encrypted and signed information. If you make changes, the key history for that user is lost and you must cancel and restart the export operation.

---

- 3 If required, change any information (such as incorrect distinguished names) in the file, or add any information (such as groups) to file.

For information about bulk commands, see [“Bulk commands reference” on page 651](#).

---

**Attention:** Information between curly braces are signed and encrypted key history. Do not add, delete, or change any characters between curly braces or you risk losing the user's key history.

---

- 4 Save the import file.
- 5 Process the import file (see [“Processing bulk files” on page 280](#)).

You have now imported users into the new CA. The number of available Entrust licenses decreases by one for each imported user.

After you import a user, you can help the user log in to the new CA. See [“Logging in to the new CA” on page 216](#).

## Logging in to the new CA

While Entrust PKI administrators at the old and new CAs are moving a user, the user can continue to work on the old CA so that their work is not interrupted by the move. The user can log in to the new CA after the Entrust PKI administrator imports the user into the new CA.

The process of logging in to the new CA depends on the user's Security Manager client application, the version of Security Manager running on the CAs, and the type of trust established between the CAs.

For Security Provider users moving between cross-certified CAs running Security Manager, the move is handled automatically, and the user can just log in. For Security Provider users moving between CAs that do not trust each other, the new CA must recover the user's keys.

---

**Note:** For Security Provider users, you must complete the export at the old CA before they can log in at the new CA. (See [“Completing user export operations” on page 217](#).)

---

Users of other client applications require an updated `entrust.ini` file that points to the new CA before logging in for the first time. (See [“Handling user information” on page 213](#).) In these cases, the user must replace the `entrust.ini` file with the edited file from the old CA. Once a user receives the updated `entrust.ini` file, the user can then log in to the new CA.

When logging in to the new CA, users must recover their Entrust profile. If the `MovingDomain=1` flag was included in the user's `entrust.ini` file, the user's profile



recovers automatically when the user logs in to the CA. (See [“Handling user information” on page 213](#).) Otherwise, the user must perform manual key recovery by entering the activation codes (generated when the user was imported). See [“Recovering the Entrust profile manually” on page 217](#).

---

**Note:** After the user has successfully logged in to the new CA, you must set the the `MovingDomain` flag in the user's `entrust.ini` file back to 0.

---

## Recovering the Entrust profile manually

The following users must recover their Entrust profile manually when logging in to their new CA for the first time:

- Entrust Intelligence Desktop Solutions or Entrust Intelligence Security Provider users whose most recent verification certificate is expired or revoked
- Entrust Intelligence Security Provider users who are moving from or to a CA on Security Manager 6.x or earlier
- Entrust Intelligence Desktop Solutions 4.x or earlier users
- users of any other Entrust client applications
- users who moved between CAs that are not cross-certified

---

**Note:** During profile recovery, users must provide activation codes (an authorization code and reference number). The activation codes generate when the user is imported.

---

After recovering the profile, the user is logged in to the new CA. A message appears indicating that the user has changed CAs.

## Completing user export operations

For Security Provider users, you must complete the export at the old CA before the user can log in at the new CA.

For users of other Entrust applications, the Entrust PKI administrator at the new CA informs the Entrust PKI administrator at the old CA that the user has been imported when a user has successfully logged in to the new CA. At this time, the Entrust PKI administrator at the old CA completes the user export operation. When the Entrust PKI administrator completes exporting the user, Security Manager removes the user's certificates from the directory and the license count increases by one. The user can no longer log in to the old CA and other users cannot access the user's certificates in the old CA.

---

**Note:** Because a deactivated user is not able to work in a CA, users who are deactivated when you export them are automatically in the `Export` state. You do not need to complete the export operation.

---

You can only complete a user export operation when the user is in the `Export Hold` state.

### To complete the export operation

- 1** Log in to Security Manager Administration as an Entrust PKI administrator (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Find the user whose properties you want to change (see [“Finding users” on page 134](#)).
- 3** Right-click the user and click **Complete Export** in the pop-up menu.
- 4** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears. The number of available Entrust licenses at the old CA increases by one for every exported user.

After you have completed the export, you may want to archive a user to remove the user’s information from the Security Manager database. For more information, see [“Archiving and retrieving users” on page 183](#).

# Configuring user properties

Entrust PKI administrators with sufficient permissions can view and change the properties for users at any time. The user properties that Entrust PKI administrators can view or change depends on the administrator's role and the state of the user.

You might change a user's properties for any number of reasons. For example, if a contract employee becomes a full-time employee, you need to change the date on which the employee's keys expire. As a contractor, this person's certificates were not updated before expiring, and the certificate expiration dates matched the expiration date of the person's contract. As a full-time employee, this person's properties must change so that the keys are updated automatically before they expire.

For details about configuring subjectAltName values for users, see ["Configuring subjectAltName values" on page 249](#). For details about configuring attribute certificates for users, see ["Administering attribute certificates" on page 635](#). For details about activation codes, see ["Managing activation codes" on page 153](#).

---

**Note:** If you change any user properties, you must update the user's certificates to apply the changes you made. For example, if you change the user's role, the user will continue to use the old role until you update the user's certificates.

To update the user's certificates, you can recover the user's keys (see ["Recovering user key pairs" on page 162](#)) or update the user's keys (see ["Updating key pairs" on page 240](#)).

---

This section contains the following topics:

- ["Configuring general user properties" on page 219](#)
- ["Configuring user certificate types" on page 221](#)
- ["Viewing and exporting user certificates" on page 223](#)
- ["Configuring user key update options" on page 230](#)
- ["Viewing DN change history" on page 237](#)
- ["Configuring user encryption and verification OIDs" on page 237](#)

## Configuring general user properties

Entrust PKI administrators with sufficient permissions can view and modify a user's general properties. You can view the user's distinguished name (DN), current and previous user state, and the dates when the user was added and activated. You can view and change the role assigned to the user, and the user's group affiliation.

## To view and change general user properties

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Find the user whose properties you want to view or modify (see [“Finding users” on page 134](#)).
- 3 Select the user and then select **Users > Selected User > Properties**.

The **User Properties** dialog box appears.

The screenshot shows the 'User Properties' dialog box for a user named 'Sample User'. The dialog has several tabs: 'General', 'SubjectAltName', 'Certificate Info', 'Certificate List', 'Key Update Options', and 'DN Cr'. The 'General' tab is selected. It contains the following fields and controls:

- User DN:** A text field containing 'cn=Sample User, ou=ca, c=MM'.
- User role:** A drop-down menu currently showing 'End User'.
- User group(s):** A section with a checked box for 'All groups'. Below it are two lists: 'Available:' containing 'default' and 'Assigned:' which is empty. Between the lists are 'Add >>' and '<< Remove' buttons.
- Current state:** A text field showing 'Active'.
- Previous state:** A text field showing 'Key Recovery'.
- Added:** A text field showing '10/12/2010 9:57:33'.
- Activated:** A text field showing '10/12/2010 9:57:57'.
- A status message at the bottom: 'Pending update of key pairs...'.

At the bottom of the dialog are four buttons: 'OK', 'Cancel', 'Apply', and 'Help'.

- 4 Click the **General** tab.
- 5 To change the user's role, select a new role from the **User role** drop-down list. For more information about roles, see [“Administering roles” on page 353](#).
- 6 To change the user's group affiliation:

- To add the user to all current and future groups, select **All groups**.
- To add the user to specific groups, deselect **All groups** and add or remove groups as required.

To add a group, select a group from the **Available** list and then click **Add**. To remove a group, select a group from the **Assigned** list and then click **Remove**.

**7** Click **OK**.

**8** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

**9** If you changed any user properties, you must update the user's certificates to apply the changes you made. For example, if you changed the user's role, the user will continue to use the old role until you update the user's certificates.

To update the user's certificates, you can recover the user's keys (see [“Recovering user key pairs” on page 162](#)) or update the user's keys (see [“Updating key pairs” on page 240](#)).

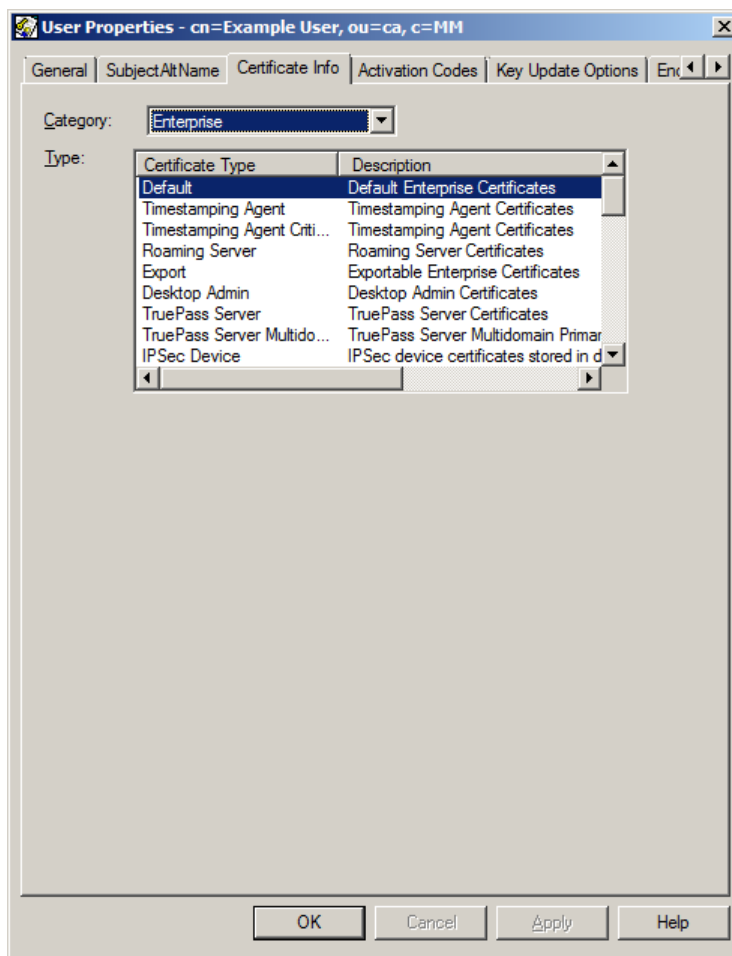
## Configuring user certificate types

Entrust PKI administrators can view and change a user's certificates. For example, you can change 1-key-pair user to a 2-key-pair user by changing the user's certificate type. You cannot change certificate categories (from Enterprise certificates to Web certificates) for active users. For more information about the default certificate types included with Security Manager, see [“Predefined certificate types” on page 535](#).

Change a user's certificate types if you originally assigned the wrong certificate type to the user, or you want to change the type of certificates that the user receives. For example, change the certificate type if you want to change the user from a 1-key-pair user to a 2-key-pair user.

### To view and change certificate types

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Find the user whose properties you want to view or modify (see [“Finding users” on page 134](#)).
- 3** Select the user and then select **Users > Selected User > Properties**.  
The **User Properties** dialog box appears.



- 4** Click the **Certificate Info** tab.
- 5** To change the certificate category (from Enterprise to Web, for example), select a new certificate category from the **Category** drop-down list.  
You cannot select a new category for active users.
- 6** To change the type of certificates that the user receives, select a new certificate type from the **Type** list.
- 7** Click **OK**.
- 8** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
If the operation was successful, a success message appears.

- 9 If you changed any user properties, you must update the user's certificates to apply the changes you made. For example, if you changed the user's role, the user will continue to use the old role until you update the user's certificates.

To update the user's certificates, you can recover the user's keys (see ["Recovering user key pairs" on page 162](#)) or update the user's keys (see ["Updating key pairs" on page 240](#)).

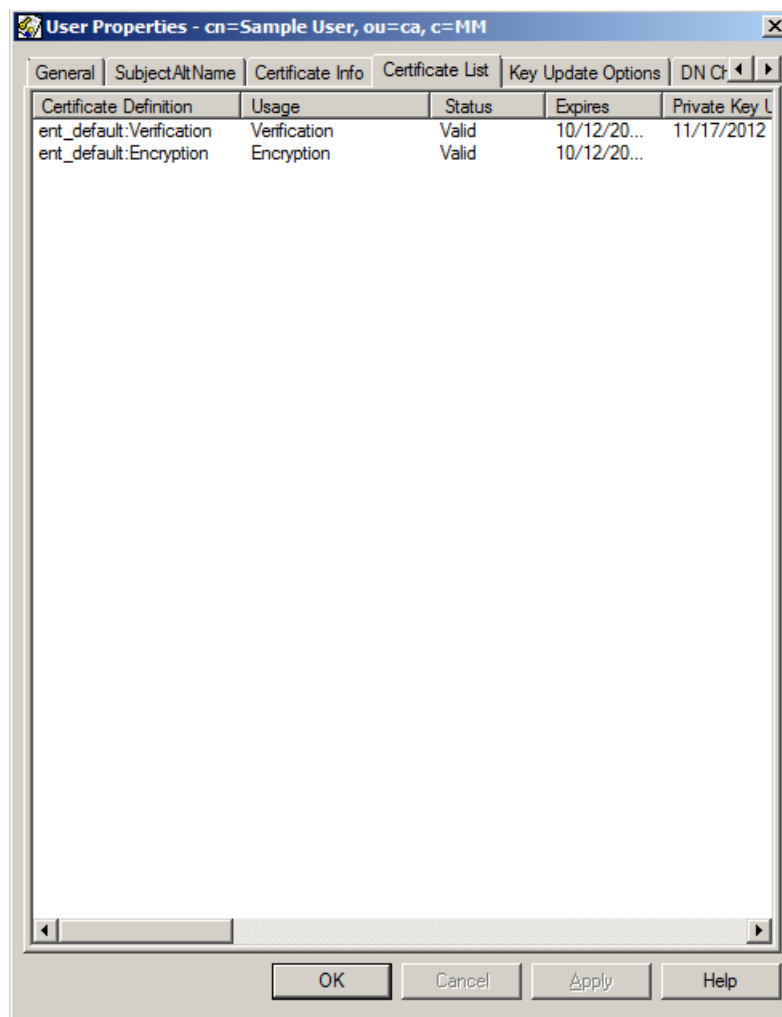
## Viewing and exporting user certificates

Entrust PKI administrators with sufficient permissions can view a user's certificates. Viewing certificates displays the certificate's usage, state, expiry date and other information. If required, you can also export the certificates.

- ["To view user certificates" on page 223](#)
- ["To export a user certificate" on page 228](#)

### To view user certificates

- 1 Log in to Security Manager Administration (see ["Logging in to Security Manager Administration" on page 46](#)).
- 2 Find the user whose properties you want to view or modify (see ["Finding users" on page 134](#)).
- 3 Select the user and then select **Users > Selected User > Properties**.  
The **User Properties** dialog box appears.

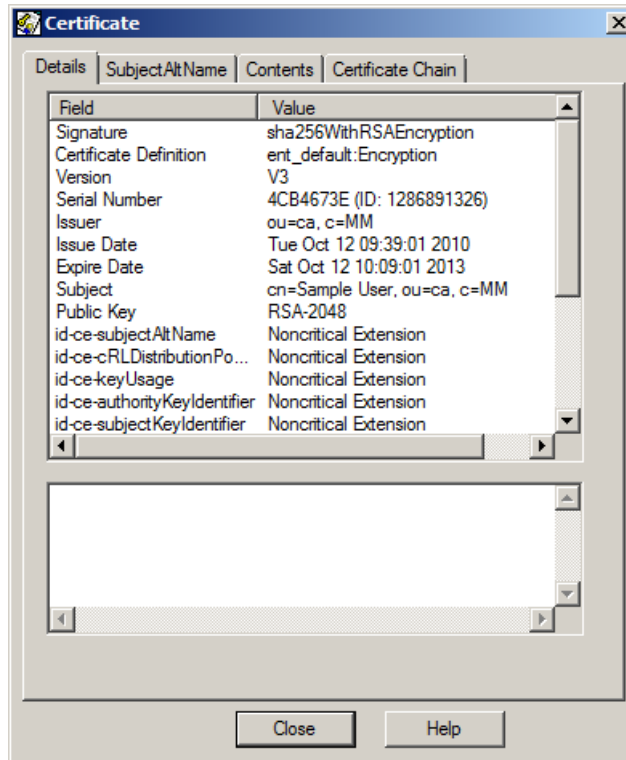


**4** Click the **Certificate List** tab.

A list of all the user's certificates displays, showing the usage, status, expiry date and other information for each certificate.

**5** Right-click the certificate you want to view and select **Certificate Contents**.  
The **Certificate** dialog box appears.



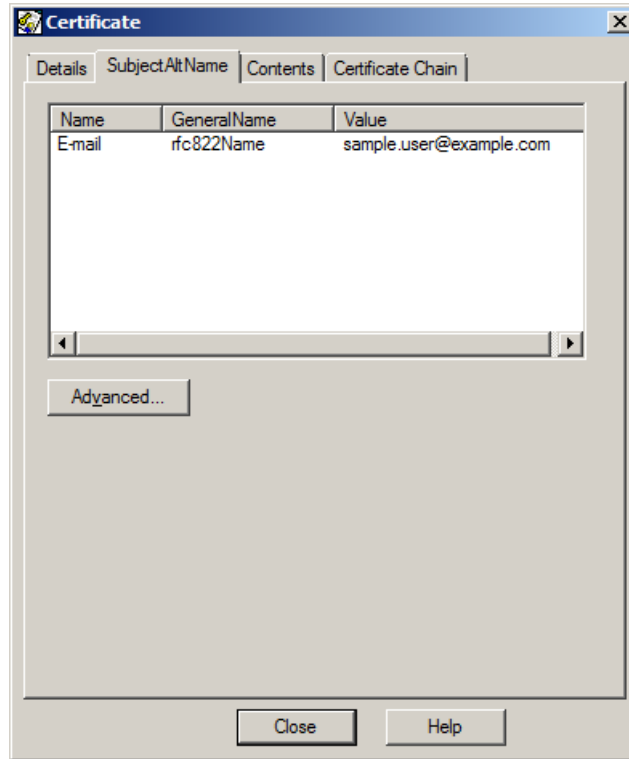


**6** Click the **Details** tab.

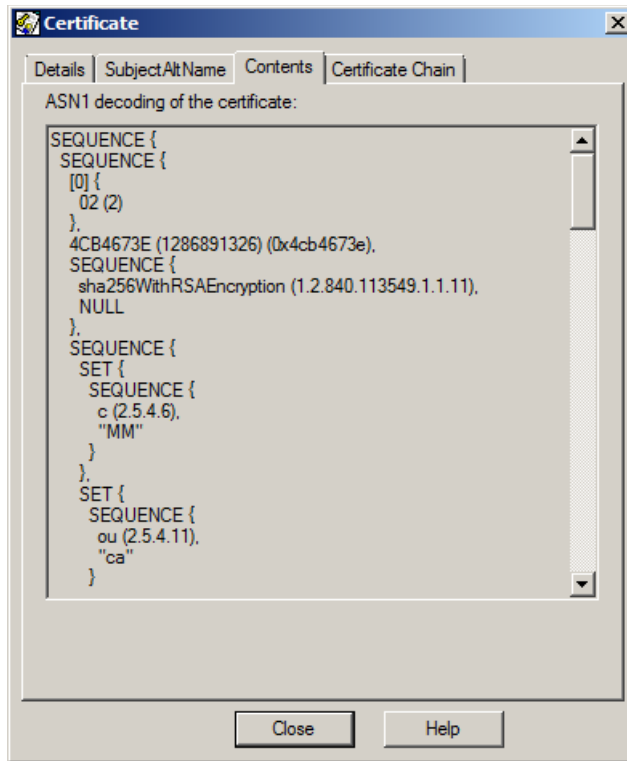
The **Details** property page contains certificate information, including the certificate definition name from the certificate specification, the serial number of the certificate, the issuing Certification Authority (CA), the encryption algorithm, the extensions in the certificate, and status information.

**7** If the certificate contains one or more `subjectAltName` extensions, the **SubjectAltName** tab becomes available. Click the **SubjectAltName** tab to view the `subjectAltName` extensions.

For more information, see [“Viewing and exporting subjectAltName component values” on page 267](#).

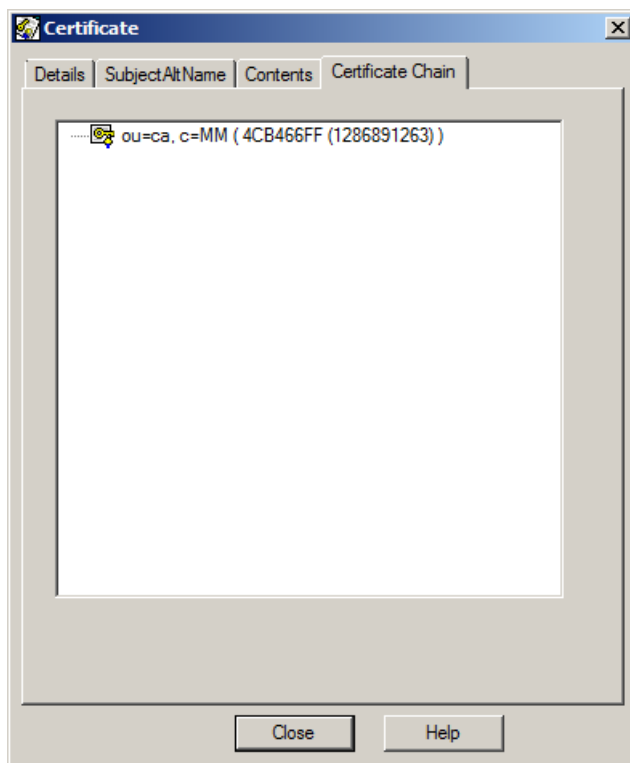


- 8 Click the **Contents** tab.  
The **Contents** property page contains an ASN.1 decoding of the certificate.



- 9 Click the **Certificate Chain** property page.

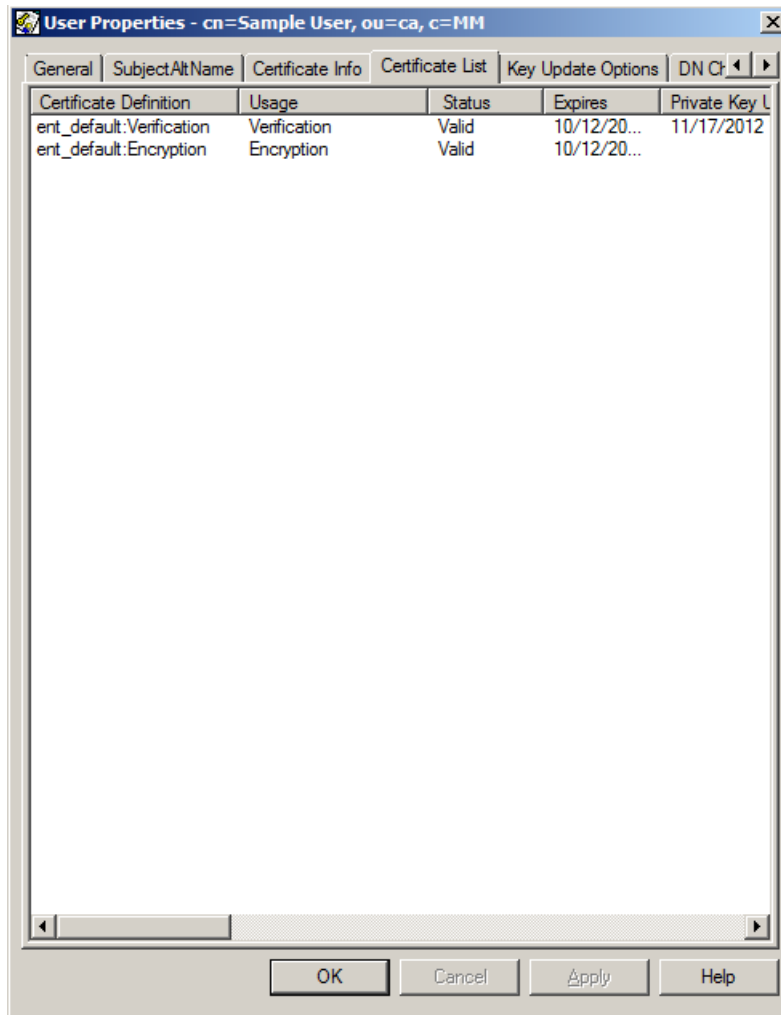
The **Certificate Chain** property page contains the chain of issuing CAs, from the CA that signed the certificate to the root CA.



**10** After viewing the properties, click **Close** to close the dialog box.

#### To export a user certificate

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Find the user whose properties you want to view or modify (see [“Finding users” on page 134](#)).
- 3** Select the user and then select **Users > Selected User > Properties**.  
The **User Properties** dialog box appears.

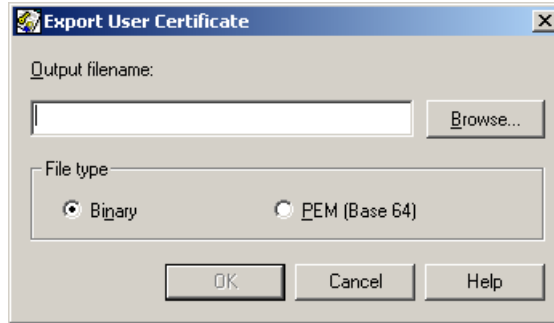


**4** Click the **Certificate List** tab.

A list of all the user's certificates displays, showing the usage, status, expiry date and other information for each certificate.

**5** Right-click the certificate you want to export and select **Export** certificate.

The **Export User Certificate** dialog box appears.



- 6** In the **Output filename** field, enter the full path and file name of the file. For example, `C:\sample_certificate.cer`.
  - 7** Click **Binary** to encode the certificate in binary format, or **PEM (Base 64)** to encode the certificate in PEM format.
  - 8** Click **OK** to export the certificate.
- If the operation was successful, a success message appears.

## Configuring user key update options

Entrust PKI administrators with sufficient permissions can view and modify the key update options for users. By default, when you add or create users, the user's keys update according to the lifetimes specified in the Security Policy (see [“Configuring options for the various certificate categories” on page 97](#)). When configuring the key update options for users, you can use the default key update policy, set custom key lifetimes, or set key expiry dates. For more information about key lifetimes, see the *Security Manager Deployment Guide*.

If you set custom key lifetimes, Security Manager automatically generates new key pairs for the user before the current key pairs expire.

---

**Note:** Security Manager automatically updates key pairs with custom lifetimes for Enterprise certificates only. Security Manager does not automatically update key pairs with custom lifetimes for Web certificates.

---

If you set fixed expiry dates, the user's key pairs expire at the specified date and time, and Security Manager does not update the user's keys.

Typically, you would only configure custom key expiry dates for users who have a planned termination date (such as contractors or temporary employees) or users working on special projects. For example, you could set the expiry date of the encryption public key and the signing private key to coincide with the contract or project end date. After the end date, the owner of the keys is no longer able to log

in to Security Manager client applications. In addition, other users are unable to encrypt files for this user.

When setting fixed expiry dates, you set the expiry date of the encryption public and signing private keys, and the expiry date of the verification public key. When setting the expiry date of the verification public key, the date should reflect the type of information that the user needs to sign. If the user is signing files that must be verifiable for a long time, the verification public key should be valid for this period of time so that the digital signature can be verified during the entire lifecycle of the documents.

For example, if your organization submits signed travel expenses online and archives them for auditing purposes, you should set the expiry date of the verification public key to at least as long as the audit period. Conversely, if the owner of the keys needs to sign information that has a short lifetime, the lifetime of the verification public key does need to extend much longer than the corresponding signing private key. Examples of such information may include email messages and attachments.

Once a verification public key expires, users can still use it to verify the signature. However, they receive a warning explaining that the key has expired and that they should not necessarily trust the signature.

---

**Note:** The verification settings for certificate lifetimes in security policy settings controls a 1-key-pair user's certificate expiry and lifetime. When the dual-usage certificate approaches expiry, the user's key pair is automatically updated (if the user policy is set for automatic key update). The previous dual-usage certificate is no longer used for encryption. The decryption function of the previous dual-usage private key, however, does not expire; it can always decrypt data that was encrypted using the previous dual-usage certificate.

---

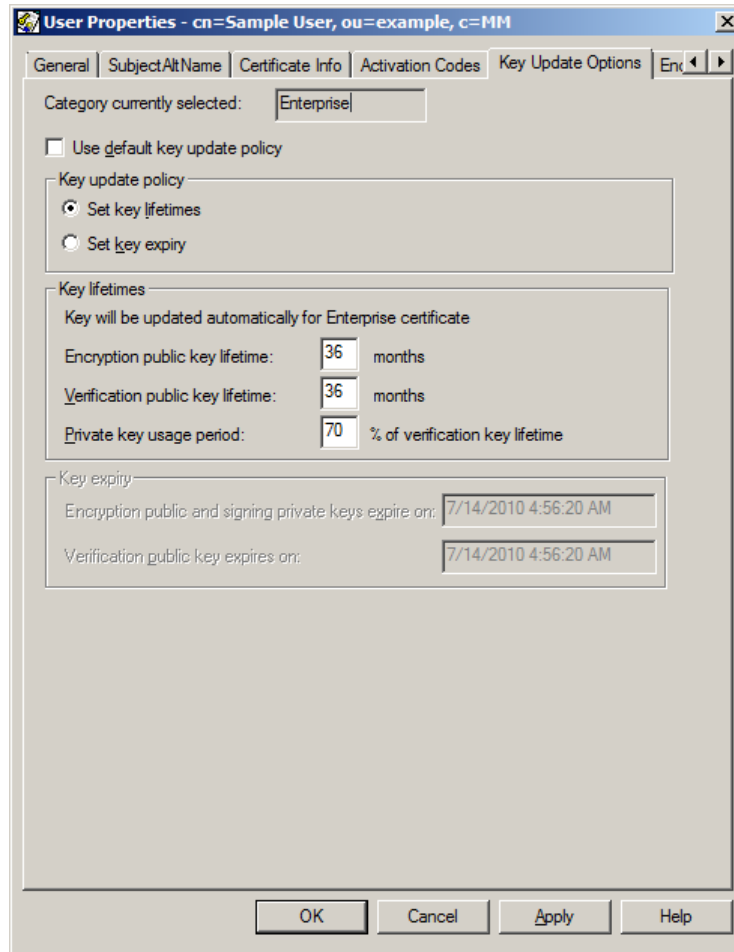
The decryption private key has no expiry date. This key is used to decrypt data encrypted with a matching encryption public key. If this key expires, you are unable to decrypt data encrypted with the encryption public key. For more information, see the *Security Manager Deployment Guide*.

- [“To set custom key update options for Enterprise certificates” on page 231](#)
- [“To set custom key update options for Web certificates” on page 234](#)

### To set custom key update options for Enterprise certificates

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Find the user whose properties you want to view or modify (see [“Finding users” on page 134](#)).
- 3 Select the user and then select **Users > Selected User > Properties**.

The **User Properties** dialog box appears.



- 4 Click the **Key Update Options** tab.
- 5 To use the key update options defined in the Security Policy, select **Use default key update policy**.
- 6 To set custom key lifetimes:
  - a Deselect **Use default key update policy**.
  - b Under **Key update policy**, click **Set key lifetimes**.
  - c Configure the following options under **Key lifetimes**:
    - In the **Encryption public key lifetime** field, enter the lifetime of the encryption public key certificate in months (from 2 to 420 months).  
If the user certificate lifetime exceeds the latest CA certificate lifetime, it truncates to the CA certificate lifetime.



- In the **Verification public key lifetime** field, enter the lifetime of the encryption public key certificate in months (from 2 to 420 months).  
If the user certificate lifetime exceeds the latest CA certificate lifetime, it truncates to the CA certificate lifetime.
- In the **Private key usage period** field, enter a percentage of the verification key lifetime (from 1 to 100).  
Security Manager will start attempting to update the user's certificates when the private key usage period reaches this percentage of the verification key lifetime.

**7** To set custom expiry dates:

- a** Deselect **Use default key update policy**.
- b** Under **Key update policy**, click **Set key expiry**.
- c** Configure the following options under **Key expiry**:
  - In the **Encryption public and signing private keys expire on** field, enter the date and time that the user's encryption public and signing keys expire. For example, the default for English (US) is `MM/DD/YYYY`. Enter the time as `hh:mm:ss` followed by `AM` or `PM`. The date and time must occur at least 12 hours into the future.  
If the specified date is past the CA certificate lifetime, Security Manager Administration returns an error.
  - In the **Verification public key expires on** field, enter the date and time that the user's verification public key expires. For example, the default for English (US) is `MM/DD/YYYY`. Enter the time as `hh:mm:ss` followed by `AM` or `PM`. The date and time must occur at least 12 hours into the future.  
If the specified date is past the CA certificate lifetime, Security Manager Administration returns an error.

**8** Click **OK**.

**9** If you set custom key lifetimes:

- a** The **Apply Changes** dialog box appears. Click **OK** to apply the changes.
- b** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
If the operation was successful, a success message appears.
- c** For an active user, update the user's keys (see [“Updating key pairs” on page 240](#)).

**10** If you set custom expiry dates:

- a** The **Apply Changes** dialog box appears. Click **OK** to apply the changes.
- b** For an active user, the **Change User** dialog box appears. This dialog box informs you that the changes you made requires that you set the user for key recovery.

Click **Yes** to apply the changes and set the user for key recovery.

- c** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

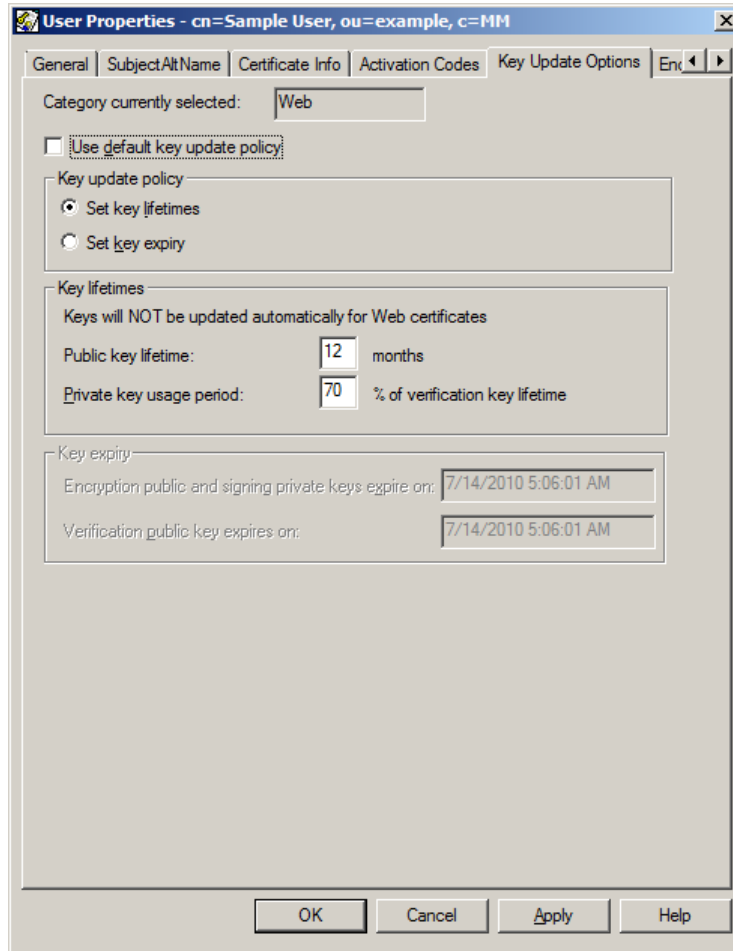
If the operation was successful, a success message appears, displaying the activation codes required to recover the user's profile.

- d** Record the activation codes and securely give them to the user (see [“Distributing activation codes” on page 153](#)).

### To set custom key update options for Web certificates

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Find the user whose properties you want to view or modify (see [“Finding users” on page 134](#)).
- 3** Select the user and then select **Users > Selected User > Properties**.

The **User Properties** dialog box appears.



- 4 Click the **Key Update Options** tab.
- 5 To use the key update options defined in the Security Policy, select **Use default key update policy**.
- 6 To set custom key lifetimes:
  - a Deselect **Use default key update policy**.
  - b Under **Key update policy**, click **Set key lifetimes**.
  - c Configure the following options under **Key lifetimes**:
    - In the **Public key lifetime** field, enter the lifetime of the encryption public key certificate in months (from 2 to 420 months).  
If the user certificate lifetime exceeds the latest CA certificate lifetime, it truncates to the CA certificate lifetime.

- In the **Private key usage period** field, enter a percentage of the verification key lifetime (from 1 to 100).
- 7** To set custom expiry dates:
  - a** Deselect **Use default key update policy**.
  - b** Under **Key update policy**, click **Set key expiry**.
  - c** Configure the following options under **Key expiry**:
    - In the **Encryption public and signing private keys expire on** field, enter the date and time that the user's encryption public and signing keys expire. For example, the default for English (US) is `MM/DD/YYYY`. Enter the time as `hh:mm:ss` followed by `AM` or `PM`. The date and time must occur at least 12 hours into the future.  
If the specified date is past the CA certificate lifetime, Security Manager Administration returns an error.
    - In the **Verification public key expires on** field, enter the date and time that the user's encryption public and signing keys expire. For example, the default for English (US) is `MM/DD/YYYY`. Enter the time as `hh:mm:ss` followed by `AM` or `PM`. The date and time must occur at least 12 hours into the future.  
If the specified date is past the CA certificate lifetime, Security Manager Administration returns an error.
- 8** Click **OK**.
- 9** If you set custom key lifetimes:
  - a** The **Apply Changes** dialog box appears. Click **OK** to apply the changes.
  - b** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
If the operation was successful, a success message appears.
  - c** For an active user, update the user's keys (see [“Updating key pairs” on page 240](#)).
- 10** If you set custom expiry dates:
  - a** The **Apply Changes** dialog box appears. Click **OK** to apply the changes.
  - b** For an active user, the **Change User** dialog box appears. This dialog box informs you that the changes you made requires that you set the user for key recovery.  
Click **Yes** to apply the changes and set the user for key recovery.
  - c** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
If the operation was successful, a success message appears, displaying the activation codes required to recover the user's profile.

- d Record the activation codes and securely give them to the user (see [“Distributing activation codes” on page 153](#)).

## Viewing DN change history

Security Manager keeps a record of all the distinguished name (DN) changes users have experienced as well as the date when the changes occurred. You can view the history of DN changes for users in Security Manager.

Users in the Added or Imported state do not have a DN change history. Security Manager does not record DN changes for added or imported users.

### To view DN change history

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Find the user whose properties you want to view or modify (see [“Finding users” on page 134](#)).
- 3 Select the user and then select **Users > Selected User > Properties**.  
The **User Properties** dialog box appears.
- 4 Click the **DN Changes** tab.  
A list of all the user’s certificates displays, showing the usage, status, expiry date and other information for each certificate.

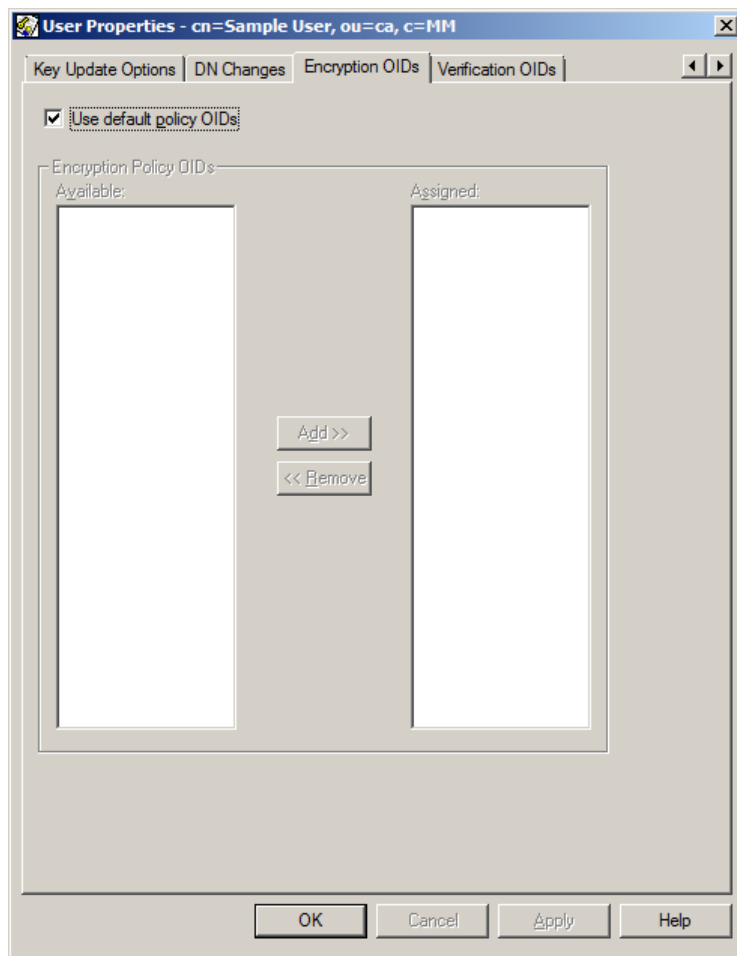
## Configuring user encryption and verification OIDs

Encryption and verification policies are represented using object identifiers (OIDs). By default, users receive the encryption and verification OIDs defined in the Security Policy (see [“Configuring the encryption and verification OIDs” on page 90](#)).

If required, you can configure users to receive a custom set of encryption and verification OIDs. The available OIDs you can choose from must already exist. See [“Adding OIDs” on page 90](#) for information about creating OIDs. You can only assign a maximum of 10 encryption and 10 verification OIDs to users.

### To modify the encryption or verification OIDs that users receive

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Find the user whose properties you want to view or modify (see [“Finding users” on page 134](#)).
- 3 Select the user and then select **Users > Selected User > Properties**.  
The **User Properties** dialog box appears.



- 4 To customize the encryption OIDs that the user receives, click the **Encryption OIDs** tab. To customize the encryption OIDs that the user receives, click the **Verification OIDs** tab.
- 5 To use the default policy OIDs defined in the Security Policy, select **Use default policy OIDs**.
- 6 To use a custom list of OIDs, deselect **Use default policy OIDs** and then configure the OIDs the user receives as follows:
  - To add an OID, select the OID in the **Available** list and then click **Add**.  
You can add a maximum of 10 OIDs to the **Assigned** list.
  - To remove an OID, select the OID in the **Assigned** list and then click **Remove**.

- 7** Click **OK**.
- 8** If prompted to authorize the operation, authorize the operation. See ["Authorizing sensitive operations" on page 52](#).  
If the operation was successful, a success message appears, displaying the activation codes required to recover the user's profile.
- 9** If you changed any user properties, you must update the user's certificates to apply the changes you made. For example, if you changed the user's role, the user will continue to use the old role until you update the user's certificates.  
To update the user's certificates, you can recover the user's keys (see ["Recovering user key pairs" on page 162](#)) or update the user's keys (see ["Updating key pairs" on page 240](#)).

# Updating key pairs

At times you may want to change a user's properties and update the user's keys to ensure that changes are recognized immediately by Security Manager. For example, you may need to add email addresses to your users' certificates. You can do this and have these new properties recognized by Security Manager without having to recover these users. By updating keys, the user's profile is automatically updated with a new signing key pair the next time the user logs in to a Security Manager client application.

Security Manager can also automatically update key pairs. If users have automatic key update, a Security Manager client application automatically generates new key pairs before the old certificates expire. The feature explained in this section is useful when you need to register a change to a user immediately. For example, you may want to update the keys immediately when you revoke a user's certificate.

There are situations when a PKI administrator cannot update a user's key pairs. If you want to update a user's key pairs, all of the following must be true:

- The user is a profile user (a user who logs into the profile using an Entrust application).
- Security Manager generates the user's key pairs.

If the user's keys are generated by a Security Manager client application on the user's computer, you cannot update the user's key pairs using Security Manager Administration.

- The Security Manager database stores back ups of decryption private keys. Decryption private keys are backed up unless you have disabled key backup.
- The user properties allow key updates, meaning that the user's certificates have a lifetime instead of an expiry date.

See ["Configuring user key update options" on page 230](#) for details about key lifetimes and expiry dates.

- The user is in the Active or Key Recovery state (see ["User states" on page 131](#)).
- If the user is in the Active state and the user's last verification certificate is revoked, then the user must be put into the Key Recovery state to update their keys.

Since a user signs the certificate request with their signing key, key update is not possible if the verification certificate is revoked. (Note that this is true whether a user's keys update automatically or a PKI administrator updates a user's keys.)

- A new encryption certificate must not already be pending in the directory.

When a PKI administrator performs a key update, a pending encryption certificate is created and put in the directory. When the client application sees this new certificate (when the user next logs in), it contacts Security Manager



to obtain the new encryption private key. At this time it also updates the signing key. Therefore, if there is already a pending certificate in the directory (a PKI administrator performed a key update, but the user has not logged in yet), the PKI administrator cannot attempt another key update.

---

**Note:** If you have set the user encryption key to RSA-4096 or RSA-6144, the key update operation is considerably slower than for RSA-1024 or RSA-2048. See the *Security Manager Operations Guide* for more information.

---

#### To update key pairs

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Find the user whose keys you want to update (see [“Finding users” on page 134](#)).
- 3** Select the user and then select **Users > Selected User > Update Key Pairs**.
- 4** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

# Notifying client applications

V2 client applications periodically check Security Manager for any events that may have occurred to their users, such as a key update or recovery. When an event occurs, the client application works with Security Manager to implement the operation. However, if you modify the `master.certspec` file and modify a user's certificate type, the V2 client application does not recognize that a change occurred.

If the change affects existing certificate definitions, then you should update the user's key pairs to indicate that the certificates must be replaced. For more information about updating a user's key pairs, see [“Updating key pairs” on page 240](#).

If the change only adds a new certificate definition and the user does not need replacement certificates for the existing certificate definitions, then you should notify the client. Notifying the client indicates to the V2 client application that the user requires a new key pair while maintaining the existing certificates.

---

**Note:** When you notify a client application, Security Manager does not contact the V2 client application. Notifying a client application allows the client application to recognize that changes occurred the next time the application checks Security Manager.

---

For more information about the `master.certspec` file and customizing certificates, see [“Customizing certificates” on page 525](#).

## To notify a client application

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Find the user whose keys you want to update. See [“Finding users” on page 134](#).
- 3 Select the user and then select **Users > Selected User > Notify Client**.
- 4 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

# Changing user profiles

Whenever an Entrust Entelligence-based client application makes a change to the user's digital ID that results in the publication of a new certificate, it sets an entry in the digital ID called `CertificatePublicationPending`, which is the time at which the client started waiting for the certificate to be published.

When the client detects that the certificate is published in the directory, it sets the `CertificatePublicationPending` flag to 0.

If the client sees that the certificate is not in the directory, it does the following:

- If the `CertificatePublicationPending` flag is 0, it contacts Security Manager and tries to change the user's distinguished name (DN).
- If the `CertificatePublicationPending` flag is not 0, it checks to see if a timeout period has passed (with respect to the time specified by the `CertificatePublicationPending` flag).

By default, the timeout period is 24 hours, but this can be overridden by adding the entry `CertificatePublicationTimeout` to the `[Entrust Settings]` section of the `entrust.ini` file. If the timeout period has not passed, the client does not contact Security Manager and continues with the login. If the timeout period has passed, the client contacts Security Manager and tries to change the user's DN.

The certificate is missing from the directory if the user is deactivated or during a DN change. In both situations, the client may reach the point where it sends a change DN request to Security Manager (depending on the timeout status). If the user is deactivated, Security Manager returns an error to inform the client of deactivation.

# Allowing profile export

Entrust Entelligence Desktop Solutions allows a user to export a public encryption certificate to the KEY or P7C format. This allows people outside of the user's domain and S/MIME-enabled users to encrypt files for the user. Users can also export their private keys for use with applications that are not Entrust desktop by synchronizing their profiles with Microsoft Cryptographic API (CAPI) or exporting to PKCS#12.

---

**Note:** This section does not apply to Entrust Entelligence Security Provider. If you want to configure export from CAPI to other applications for a Security Provider user, see the Security Provider documentation.

---

Exporting to PKCS #12 format allows Entrust PKI users to export their digital identities, including private keys, certificates, miscellaneous secrets, and extensions to a password-protected PKCS#12 file. This enables users to conduct secure transactions with other people who also support the PKCS#12 standard, regardless of their PKI vendor.

Synchronizing with CAPI means that a user's certificates and private keys are imported into the Microsoft CAPI store for use with CAPI-enabled applications (such as Microsoft Outlook Express and Microsoft Internet Explorer).

---

**Note:** CAPI profile export is supported for users in subordinate CAs, but is not supported for users in cross-certified CAs. Cross-certificates are not exported when exporting to CAPI.

---

This section shows you how to allow users to export their profiles to PKCS #12 format and to CAPI. For more information, see the Desktop Solutions documentation.

Topics in this section:

- [“Overview of allowing profile export” on page 245](#)
- [“Creating or modifying a user policy to allow profile export” on page 246](#)
- [“Assigning the user policy to a role” on page 246](#)
- [“Creating or modifying users to allow profile export” on page 246](#)

## Overview of allowing profile export

Before a user can export a profile:

- The user's encryption certificate, verification certificate, or both certificates must contain an extension allowing profile export.  
(If the user is a 1-key-pair user, the user's dual-usage certificate must contain the extension.)
- The user's policy certificate must allow PKCS #12 export, CAPI export, or both.

---

**Note:** Users who use hardware tokens cannot export their profiles in PKCS #12 format.

---

To allow users to export their profile, an Entrust PKI administrator must:

- 1 Create or modify a user policy to allow profile export to PKCS #12 format, CAPI, or both (see [“Creating or modifying a user policy to allow profile export” on page 246](#)).
- 2 Assign the user policy to a role (see [“Assigning the user policy to a role” on page 246](#)).
- 3 Assign the role and the Export certificate type to the users (see [“Creating or modifying users to allow profile export” on page 246](#)).

Once these tasks are complete, all new users created with profile export capability have the profile export extension included in their user certificates when they create their profile.

Depending on which keys the certificate extension allows the user to export, the user with profile export capability is able to export the private signing key, the private decryption key, or both keys. If the user is exporting to CAPI, the encryption public key certificate, verification public key certificate, or both certificates are exported along with the corresponding private key.

If an Entrust PKI administrator modifies active users to allow profile export, you must update the user's keys (see [“Updating key pairs” on page 240](#)) or recover the user's keys (see [“Recovering user key pairs” on page 162](#)).

## Roaming users and CAPI profile export

CAPI profile export is supported for roaming users, but is not recommended in all situations. Roaming typically mandates a zero footprint on the client machine, while CAPI is based on local storage of certificates and keys. CAPI export executes whenever an Entrust PKI user logs in using a roaming profile. The certificates and keys imported into the CAPI store remain once the user has logged out.

When deciding whether to enable CAPI export for a roaming user, consider the following:

- If the roaming user consistently logs in to a personal Windows account on one or more machines, and the operating system is Windows 2000, then it is acceptable to allow CAPI export for the user.
- If roaming users are sharing Windows accounts (in a kiosk situation or on home computers) or are using operating systems other than Windows 2000, it is not recommend that you allow CAPI export.

## Creating or modifying a user policy to allow profile export

To allow users to export their profiles, an Entrust PKI administrator with sufficient permissions must create or modify a user policy to allow profile export.

You can create a user policy that allows only PKCS#12 profile export, only CAPI profile export, or both PKCS#12 and CAPI profile export.

The **Allow PKCS#12 Export** and **Minimum PKCS#12 Hash Count** client policy settings control PKCS#12 profile export. The **Enable CAPI Synchronization**, **Unprotected CAPI key storage?**, and **Private key export from CAPI?** client policy settings control CAPI profile export.

For more information about creating and modifying user policies and client policy settings, see [“Administering user policies” on page 391](#).

## Assigning the user policy to a role

After creating or modifying a user policy to allow profile export (see [“Creating or modifying a user policy to allow profile export” on page 246](#)), you must assign that policy to a role. To assign the policy to role, you can create a new role or modify an existing role.

For information about creating or modifying roles, see [“Administering roles” on page 353](#).

## Creating or modifying users to allow profile export

After assigning the user policy to a role (see [“Assigning the user policy to a role” on page 246](#)), you must assign that role to users. You can assign that role when adding or creating new users, or you can assign the role to existing users.

You must also assign the Export certificate type to the users. For more information about the Export certificate type, see [“Predefined certificate types” on page 535](#).

If you modify active users to allow profile export, you must update the user's keys (see [“Updating key pairs” on page 240](#)) or recover the user's keys (see [“Recovering user key pairs” on page 162](#)).

# Converting V2 users to V1 users

If a user wants to change from using a V2 application (such as Security Provider) to an application that uses V1 digital IDs, the following high-level steps must be carried out:

- 1 Using Security Manager Administration, a PKI administrator converts the user from V2 to V1.
- 2 In most cases, a key recover is required to complete the conversion. The PKI administrator sets the user for key recovery.

When their status has been changed in Security Manager, the Entrust PKI administrator can provide the user with the reference number and authorization code so that they can recover their profile.

## To convert a V2 user to a V1 user

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Find the user whose requires a profile (see [“Finding users” on page 134](#)).
- 3 Select **Users > Selected User > Convert to V1 Protocol Version**.
- 4 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

- 5 In most cases, a key recover is required to complete the conversion. Recover the user's key pairs (see [“Recovering user key pairs” on page 162](#)).

If a key recover is not practical, contact Entrust Customer Support (see [“Obtaining technical assistance” on page 28](#)).

- 6 Recovering the user's key pairs generates new activation codes for the user. You must give these new activation codes to the user.

The user needs these codes to complete the reactivation operation. When the user next logs in to the Security Manager client application, they must enter these codes to reactivate their profile.

Activation codes are retained until the user recovers the profile. To view these activation codes, see [“Viewing activation codes and expiry dates” on page 155](#).

Conversion of a digital ID's data to V1-key-pair means that existing V2-key-pair Entrust security stores can no longer be used with a V2-key-pair client. If you want to convert back to V2-key-pair again, you must recover the user.





## Configuring subjectAltName values

The `subjectAltName` extension allows you to add and modify alternative user identities to a certificate. This section describes how you can configure the `subjectAltName` extension for a particular user in Security Manager.

This chapter includes the following sections:

- [“Using the subjectAltName extension” on page 250](#)
- [“SubjectAltName components” on page 251](#)
- [“Configuring auto-population of the subjectAltName from the directory” on page 257](#)
- [“Adding, modifying, or deleting subjectAltName component values” on page 260](#)
- [“Updating subjectAltName component values from the directory” on page 263](#)
- [“Viewing and exporting subjectAltName component values” on page 267](#)
- [“Excluding the subjectAltName from certificate definitions” on page 270](#)
- [“Setting the criticality of the subjectAltName extension” on page 273](#)

# Using the subjectAltName extension

The `subjectAltName` extension allows you to add alternative user identities to a certificate so that applications can perform additional checks to ensure that the certificate is valid for that particular user.

For example, if you are creating certificates for use in S/MIME email applications, you can use the `subjectAltName` extension to store the unique email address of the user. Also, to use TLS, your Web server certificate may need the Fully Qualified Domain Name (FQDN) of the server in the `subjectAltName` extension. This allows browsers to perform an additional check on the certificate.

Security Manager adds `subjectAltName` component values to its database, and, when creating the certificates, includes them in the certificate information. Security Manager can add them in the following circumstances:

- When the user is added using attributes from the directory.

If you want to set up auto-population from a directory, see [“Configuring auto-population of the subjectAltName from the directory” on page 257](#).

Should the directory not already contain the values, you can customize the **Naming** tab in the **New Users** dialog box so that an administrator manually enters the information when creating the user. To do so, see [“Modifying the user template file and user types” on page 447](#).

- When the user entry is modified using Security Manager Administration.

In Security Manager Administration, you can add or modify `subjectAltName` component values after a user is created—either individually or in bulk. You then need to update the user’s certificates by updating the user’s key pairs.

If you are adding or modifying the `subjectAltName` component values after a user is created, see [“Adding, modifying, or deleting subjectAltName component values” on page 260](#). You can choose to refresh the values from changes made to the directory, provided that you have set up auto-population. To do so, see [“Updating subjectAltName component values from the directory” on page 263](#).

For information on adding the `subjectAltName` component values in bulk, see [“Adding users in bulk to Security Manager” on page 288](#).

# SubjectAltName components

Table 17 describes the SubjectAltName components supported by Security Manager.

**Table 17:** SubjectAltName component description

Component	General Name Type	Description
E-mail	rfc822Name	<p>An email address.</p> <p>Used when creating digital IDs for use with S/MIME compatible email clients, such as Entrust Entelligence Messaging Server, or Microsoft Outlook. Though not strictly required, adding email addresses allows for additional verification during the encryption process.</p> <p>For more information, see <a href="#">"Email addresses in Security Manager" on page 132.</a></p> <p>Example: <code>alice.grey@example.com</code></p> <p>You can have any number of email addresses in the <code>subjectAltName</code>.</p> <p>E-mail is mapped to the <code>mail</code> directory attribute by default.</p>
User Principal Name (UPN)	userPrincipalName	<p>The Internet-style login name of the user, specifically for use with Active Directory and the Microsoft Windows Smart card Logon feature.</p> <p>Example: <code>alice.grey@example.com</code></p> <p>You can have any number of UPNs in the <code>subjectAltName</code>.</p> <p>User Principal Name is mapped to the <code>userPrincipalName</code> directory attribute in Active Directory by default.</p>
IP Address	iPAddress	<p>The valid IP address (IPv4 or IPv6) of the computer, machine or device for which you are creating certificates.</p> <p>Used when setting up TLS and the URL uses the IP address rather than a FQDN.</p> <p>Example: <code>10.3.142.43</code></p> <p>You can have any number of IP addresses in the <code>subjectAltName</code>.</p>

**Table 17:** SubjectAltName component description (continued)

Component	General Name Type	Description
MsGUID	otherName	<p>The Microsoft domain controller Globally Unique Identifier (GUID), which is stored in an Active Directory attribute.</p> <p>Used when creating computer certificates for domain controllers and the Security Manager directory is Active Directory, Active Directory Application Mode (ADAM), or Active Directory Lightweight Directory Services (AD LDS).</p> <p>Security Manager requires ASCII-encoded HEX code syntax.</p> <p>Example: E99E82D5DEED11D2B15C00C04F5CB503</p> <p>You can have any number of GUIDs in the <code>subjectAltName</code>.</p> <p>MsGUID is mapped to the <code>objectguid</code> directory attribute in Active Directory by default.</p>
FASC-N	otherName	<p>The Personal Identity Verification (PIV) Federal Agency Smart Credential Number (FASC-N).</p> <p>Used for the FASC-N of United States federal government employees and contractors.</p> <p>Security Manager requires ASCII-encoded HEX code syntax.</p> <p><b>Note:</b> You can only have one FASC-N value in a <code>subjectAltName</code> extension. The FASC-N only needs to be in the certificates used for PIV authentication (as specified in FIPS 201).</p> <p>For more information, refer to <a href="http://csrc.nist.gov/piv-program/">http://csrc.nist.gov/piv-program/</a>.</p> <p>Example: D0439458210C2C19A0846D83685A1082108CE73984108CA3FC</p> <p><b>Note:</b> When added, Security Manager converts this value to upper case.</p>

**Table 17:** SubjectAltName component description (continued)

Component	General Name Type	Description
Permanent Identifier	otherName	<p>The unique name of an individual assigned by the organization. If included in a certificate, the third-party application can identify an individual even if the certificate, DN, or other attribute changes. This name is usually unique across the entire hierarchy of CAs in your organization.</p> <p>Use if the DNs of your users change frequently or if your users have multiple role identities. The permanent identifier allows you to maintain access control, nonrepudiation, and audit records on these users.</p> <p>There are two optional parts to the permanent identifier:</p> <ul style="list-style-type: none"><li>• a unique UTF-8 string identifier</li><li>• an OID value that identifies the Assigning Authority (CA) and the type of the permanent identifier.</li></ul> <p>For more information, see <a href="http://ftp.rfc-editor.org/in-notes/rfc4043.txt">http://ftp.rfc-editor.org/in-notes/rfc4043.txt</a>.</p> <p>Example:</p> <p>Identifier: 34abc71654 OID: 2.4.6.8</p> <p>You can have any number of permanent identifiers in the <code>subjectAltName</code>.</p> <p><b>Note:</b> You cannot auto-populate values for the permanent identifier component.</p>

**Table 17:** SubjectAltName component description (continued)

Component	General Name Type	Description
DNS Name	dNSName	<p>The DNS value of the computer, machine or device for which you are creating certificates.</p> <p>If you are enabling Microsoft Windows client authentication and are creating certificates for computers or other devices, you may need to set the DNS value in the <code>subjectAltName</code> extension.</p> <p>SSL 3.0 and TLS 1.0 require that a secure Web site's name match either the <code>dNSName</code> contained in the Web server certificate <code>subjectAltNames</code> extension or the subject common name.</p> <p>The DNS value can be the computer name, the fully qualified computer name (FQDN), or an IP address.</p> <p>Example:</p> <p>domain.example.com</p> <p>You can have any number of DNS entries in the <code>subjectAltName</code>.</p>
X400 Address	x400Address	<p>The ASN.1 DER encoded value of the X.400 address (ORAddress).</p> <p>Used with S/MIME certificates to check whether the originator's X.400 address matches the X.400 address in the <code>subjectAltName</code> extension. However, most S/MIME applications check the email address only.</p> <p><b>Note:</b> Security Manager converts this value to upper case.</p> <p>You can have any number of X.400 addresses in the <code>subjectAltName</code>.</p>

**Table 17:** SubjectAltName component description (continued)

Component	General Name Type	Description
Directory Name (DN)	directoryName	<p>The distinguished name (DN) of the directory in RFC 2253 format.</p> <p>Used with an LDAP directory to enable SSL over LDAP.</p> <p>For example: cn=directory,o=yourCompany,c=CA</p> <p><b>Note:</b> When added using Security Manager Administration, any spaces you add between RDNs are removed by Security Manager. For example, if you add cn=directory, o=yourCompany, c=CA, it becomes cn=directory,o=yourCompany,c=CA.</p> <p>You can have any number of directory Names in the subjectAltName.</p>
EDI Party Name	ediPartyName	<p>The ASN.1 DER encoding of the EDIPartyName.</p> <p>Used when securing communication between Electronic Data Interchange (EDI) partners.</p> <p><b>Note:</b> When added, Security Manager converts this value to upper case.</p> <p>You can have any number of EDI party names in the subjectAltName.</p>
Uniform Resource Identifier (URI)	uniformResourceIdentifier	<p>A Uniform Resource Identifier for the World-Wide Web defined in accordance with Internet RFC 1630.</p> <p>Used when setting up SSL certificates or when creating S/MIME certificates for the purposes of reference integrity.</p> <p>For example: http://domain.example.com</p> <p>You can have any number of URIs in the subjectAltName.</p>
Registered Identifier	registeredID	<p>The OID of any registered object assigned in accordance with ITU-T Rec. X.660   ISO/IEC 9834-1.</p> <p>For example: 2.3.7.8</p> <p>You can have any number of registered identifiers in the subjectAltName.</p>

**Table 17:** SubjectAltName component description (continued)

Component	General Name Type	Description
Other Name	otherName	<p>An ASCII-HEX DER-encoding of an ASN.1 otherName.</p> <p>Used when you have a unique name for an entity (that is, an unknown OID-value pair) that does not fit into the other subjectAltName component categories.</p> <p>For example, you could add an MsGUID value such as ac4b2906aad65d4fa99c4cbcb06a65d9 by DER-encoding it in the following format:</p> <pre>SEQUENCE {     OBJECTIDENTIFIER 1.3.6.1.4.1.311.25.1     [0] { OCTETSTRING ac4b2906aad65d4fa99c4cbcb06a65d9 } }</pre> <p>Once you have obtained the DER-encoded result (301f06092b0601040182371901a0120410ac4b2906aad65d4fa99c4cbcb06a65d9), you can enter it as the other name component value.</p> <p>You can have any number of other names in the subjectAltName.</p>



# Configuring auto-population of the subjectAltName from the directory

You can configure Security Manager so that when you add users, you automatically populate the user's `subjectAltName` value with information stored in directory attribute values. Security Manager puts this information in the `subjectAltName` extensions of the user's certificates when it creates the certificates.

By default, Security Manager puts the email address of the user into the `subjectAltName` E-mail component, if it exists in the `mail` directory attribute or if the administrator enters the email address in the **Email** field when creating the user. (If you store email addresses in a directory attribute other than `mail`, you need to modify Security Manager as documented in this section.)

---

**Attention:** You cannot auto-populate values for the permanent identifier `subjectAltName` component.

---

For more information on how Security Manager configures the `subjectAltName`, see [“Using the subjectAltName extension” on page 250](#).

When you modify users, you can also refresh their existing `subjectAltName` value with the directory attribute values (see [“Updating subjectAltName component values from the directory” on page 263](#)).

You can choose to define the `subjectAltName` for all end entity certificates, or for certificates within a certificate type. End entity certificates are certificates created using the Web or Enterprise certificate category. If you specify both, the definitions in the certificate type have precedence.

## To configure auto-population of the subjectAltName from the directory

- 1 Export the certificate specification file (`master.certspec`). For instructions, see [“Creating the master.certspec file” on page 541](#).
- 2 Find the correct place to add the `subjectAltName` definition using one of the following methods:

- to define the `subjectAltName` for all end entity certificates, search for the following line:

```
[SubjectAltName Attributes]
```

The default auto-population instruction for email is located underneath:

```
mail=email,0,1,1
```

- to define the `subjectAltName` for a particular certificate type, go to the end of the `[Attributes]` section and add the following line:

```
[<certificatetype> Common SubjectAltName Attributes]
```

where <certificatetype> is the certificate type name as defined in the [Certificate Types] section.

**3** Under the appropriate entry, create a new line with the following information:

```
<directoryattribute>=<SANcomponentname>,<mandatoryflag>,<allvalues>,<only_if_absent>
```

Where:

- <directoryattribute> is the name of the attribute that Security Manager should query for the subjectAltName value.
- <SANcomponentname> is the Security Manager name for the subjectAltName component that it creates. Security Manager allows both long and short names. One of:

- rfc822Name **OR** email
- ipAddress **OR** ip
- userPrincipalName **OR** UPN
- dNSName **OR** dns
- x400Address **OR** x400
- directoryName **OR** dn
- ediPartyName **OR** edi
- uniformResourceIdentifier **OR** uri
- registeredID **OR** oid
- MsGlobalUniqueId **OR** MsGUID
- PivFASC-N **OR** FASC-N

See “[SubjectAltName components](#)” on page 251 for details. You cannot use otherName.

---

**Attention:** You cannot auto-populate values for the permanent identifier subjectAltName component.

---

- <mandatoryflag> is 1 if you want the operation to fail if there is no value in the directory attribute, otherwise 0.
- <allvalues> is 1 if all values should be added, or 0 if only one value is added (that is, the first of multiple values returned by the LDAP search).
- <only\_if\_absent> is 1 if you do not want to add the directory attribute values to existing subjectAltName component values.

Set to 1 if you do not want the attribute value to auto-populate into the subjectAltName should it already contain a component entry mapped to the attribute.

For example, should the subjectAltName already contain an email component, and you have mapped the directory mail attribute to email,

Security Manager does not add `mail` attribute values from the directory to the existing email component.

This parameter affects bulk operations as well. For details, see [Step 13 on page 291](#).

---

**Note:** `<only_if_absent>` is not considered if you are refreshing from the directory after adding the user entry. See [“Updating subjectAltName component values from the directory” on page 263](#).

---

For example, `userPrincipalName=upn,1,0,0:`

- Maps the `userPrincipalName` directory attribute to the UPN component name.
- Makes it mandatory for a value to exist.
- Adds only the first value if multiple `userPrincipalName` values exist.
- Adds the attribute values even if the `subjectAltName` already contains mapped components.

**4** Add as many mappings as required.

Ensure that if you are mapping multiple attributes to the same `subjectAltName` component, multiple entries of that component are supported.

---

**Note:** Additional `subjectAltName` component values increases the size of each certificate. If you have smart card users, ensure that there is sufficient space for a larger certificate.

---

**5** Save your changes.

**6** Import the new certificate specification file. For instructions, see [“Processing changes to the master.certspec file” on page 543](#).

# Adding, modifying, or deleting subjectAltName component values

Complete the following task to add, modify, or delete the `subjectAltName` component value for an existing Security Manager user.

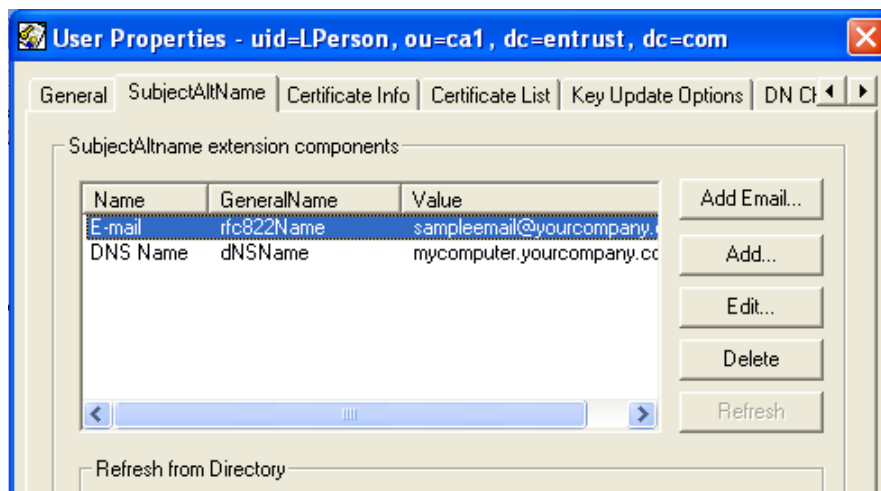
---

**Note:** To complete this task for multiple users at once, see [“To set the user’s subjectAltName component values in bulk” on page 315](#).

---

**To add, modify, or delete the subjectAltName component values of an existing user**

- 1 Log in to Security Manager Administration as an Entrust PKI administrator. See [“Logging in to Security Manager Administration” on page 46](#).
- 2 Find the user entry. For instructions, see [“Finding users” on page 134](#).
- 3 Right-click on the entry and click **Properties** to open the **User Properties** dialog box.
- 4 Click on the **SubjectAltName** tab. Entries that you have added previously appear in the list.



You can now do the following:

- add an email address ([Step 5](#))
- add other `subjectAltName` components, as listed in [“SubjectAltName components” on page 251](#) ([Step 6](#))

- if you have set up auto-population from a directory, updated the user's directory entry after creation, and want to add those values to the `subjectAltName`, see ["Updating subjectAltName component values from the directory" on page 263](#)
- modify an existing `subjectAltName` component by selecting it and clicking **Edit**.
- delete an existing `subjectAltName` component by selecting it and clicking **Delete**.

To save your changes, click **Apply**. You may need to authorize the request. To update the existing certificates, you need to update the user's key pairs. For instructions, see ["Updating key pairs" on page 240](#).

---

**Note:** When adding values, Security Manager automatically escapes special characters such as > or " using the backslash (\) when you click **OK**. If your string includes a backslash, you must prefix it with another backslash (\\) when you enter it.

---

- 5** To add an email address, click **Add Email** to enter it.

If you are entering multiple email addresses, add each one separately. You cannot add multiple email addresses at a time. For information about how Security Manager uses email addresses, see ["Email addresses in Security Manager" on page 132](#).

- 6** To add other `subjectAltName` components, click **Add**.

The **Add SubjectAltName component** dialog box opens.

- a** Under **Select component name**, select the component name matching the value you want to enter. For details on these components, see [Table 17 on page 251](#).

The **Description** and **Enter component value** fields change depending on your component selection.

- b** Enter the correct value for your `subjectAltName` extension component.

---

**Note:** Security Manager Administration does not check to ensure that the value is correct. If component values are incorrect, the certificates created for the user may fail authentication. Should this occur, you must edit the `subjectAltName` component and update the keys and certificates.

---

For example, if you are adding a DNS hostname, the **Add SubjectAltName** component dialog box appears as follows:

Name	GeneralName	Multiple values
MsGUID	otherName	Yes
FASC-N	otherName	No
Permanent Identifier	otherName	Yes
Other Name	otherName	Yes
<b>DNS Name</b>	<b>dNSName</b>	<b>Yes</b>
X400 Address	x400Address	Yes
Directory Name (DN)	directoryName	Yes
FDI Partu Name	ediPartuName	Yes

Description:

SubjectAltName dNSName component

Fields:

Text string - mandatory

Asterisks (\*) appear beside required fields.

Enter component value:

\* Text string

mycomputer@mycompany.com

OK Cancel Help

- c Click **OK**.

The new component value appears in the list under the **SubjectAltName** tab.

- d Click **Apply** in the **SubjectAltName** tab to save your changes.

Repeat this step to add as many component values as you require.

To update the existing certificates, you need to wait until the user's keys update automatically or update the key pairs manually. For instructions, see ["Updating key pairs" on page 240](#).

# Updating subjectAltName component values from the directory

Over time, you may update the user entry in the directory. If you have set up auto-population (see [“Configuring auto-population of the subjectAltName from the directory” on page 257](#)), then you can update the `subjectAltName` components by refreshing them using the updated information in the directory.

You can do so in three ways:

- replace all `subjectAltName` components with new directory values.  
Use this option if you are creating the `subjectAltName` extension only from directory attributes.
- replace values of the same kind with the updated directory values.  
For example, if you auto-populate only email addresses from the directory, replace all user email addresses with updated directory information. Use when you create the `subjectAltName` extension using values from the directory and values manually entered using Security Manager Administration ([“Adding, modifying, or deleting subjectAltName component values” on page 260](#)).
- append the updated directory values to the existing manually entered `subjectAltName` components.  
Should you have a particular value already in the `subjectAltName`, it is not updated.

Procedures in this section:

- [“To update the subjectAltName component values for a user” on page 263](#)
- [“To update the subjectAltName component values in bulk” on page 265](#)

## To update the subjectAltName component values for a user

- 1 Log in to Security Manager Administration as an Entrust PKI administrator. See [“Logging in to Security Manager Administration” on page 46](#).
- 2 Find the user entry. For instructions, see [“Finding users” on page 134](#).
- 3 Right-click on the entry and click **Properties** to open the **User Properties** dialog box.
- 4 Click on the **SubjectAltName** tab. Entries that you have added previously appear in the list.

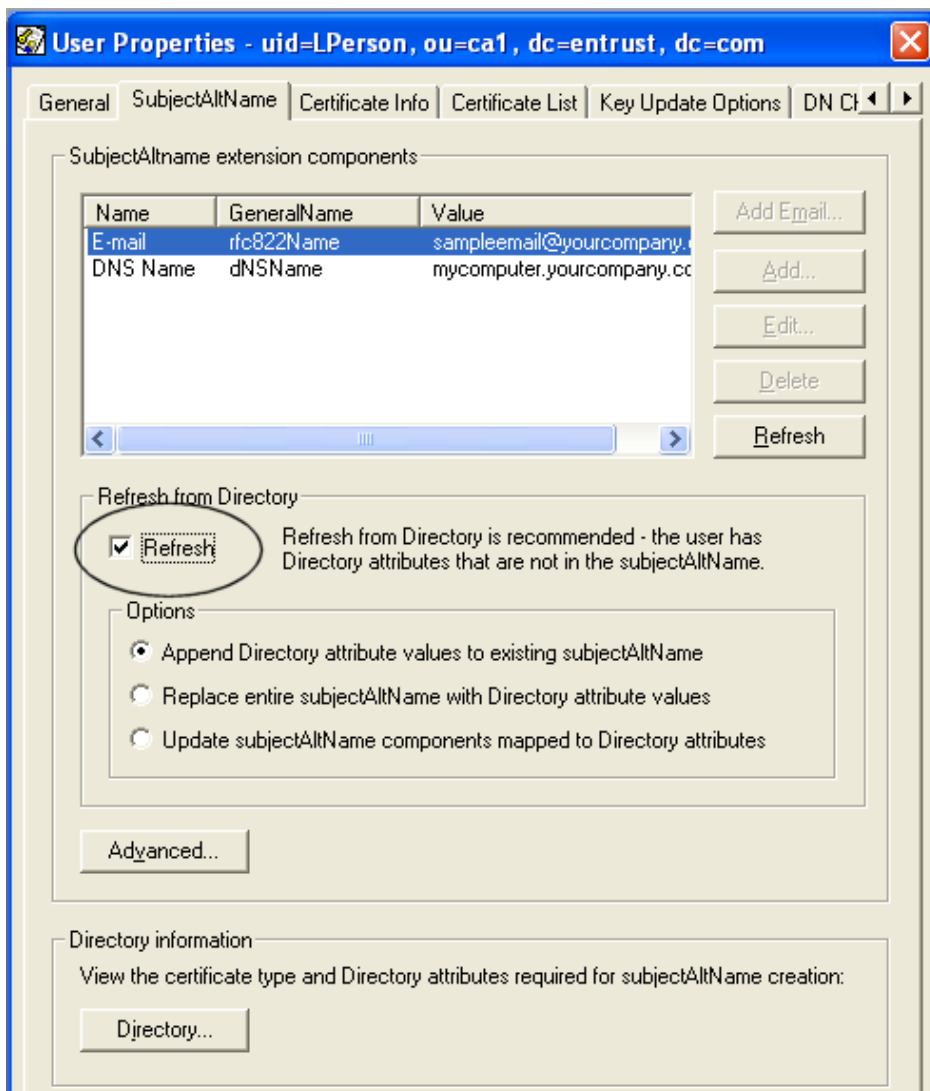
---

**Note:** To view the directory attribute values for the user, click the **Directory** button.

---

**5** Select **Refresh** under **Refresh from Directory**.

**Note:** This button is grayed out if the attribute values have not changed in the directory and you have not set up auto-population.



- 6** You have three options:
- append the new values from the directory to existing subjectAltName component values



This option adds new components and does not modify existing components or their values.

- replace the entire `subjectAltName` with directory values.

This option deletes the existing `subjectAltName` components and creates new ones based on directory values.

- update only those `subjectAltName` components that map to directory attributes

This option allows you to keep the `subjectAltName` component values that you have added through other means than auto-population. Any component that is mapped to a directory attribute is replaced with new information from the directory.

**7** Click the **Refresh** button.

A Results page appears listing the `subjectAltName` components that are affected by the directory updates.

**8** Click **Save** and authorize the request.

**9** Click **OK** to close the **User Properties** dialog box.

To update the existing certificates, you need to update the key pairs. For instructions, see [“Updating key pairs” on page 240](#).

### To update the `subjectAltName` component values in bulk

**1** Open a new file in any text editor.

**2** Identify the user you want to move by typing the following, for example:

```
user <userID> "<userDN>"
```

Where:

- `<userID>` is the label used to identify the user object in this bulk operation.
- `<userDN>` is the DN of the user. Remember to surround the user's DN with double quotation marks.

For example:

```
user z "cn=Bob Jones,ou=Chicago,dc=Company One,dc=com"
```

**3** Update the user's `subjectAltName` values using the following command:

```
user_refresh_subaltname_from_dir <userID> <refreshMode>
```

Where:

- `<userID>` is the label used to identify the user object in this bulk operation.
- `<refreshMode>` is one of:
  - `replace`. Deletes the existing `subjectAltName` components and creates new ones based on directory values.

- **update** (default). Allows you to keep the `subjectAltName` component values that you have added through other means than auto-population. Any component that is mapped to a directory attribute is replaced with new information from the directory.
- **append**. Adds new components and does not modify existing components or their values.

For example, to append new `subjectAltName` component values, type:

```
user_refresh_subaltname_from_dir z append
```

- 4** Repeat [Step 2](#) and [Step 3](#) for all the users whose `subjectAltName` component values you want to update.
- 5** Save the file with a `.entra` extension. The file can have any name (for example, `changingSAN.entra`) as long as it has the `.entra` extension.
- 6** Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).

To update the existing certificates, you need to update the key pairs. For instructions, see [“Updating users’ key pairs in bulk” on page 326](#).

# Viewing and exporting subjectAltName component values

Once you add the `subjectAltName` component values to Security Manager, they are converted and stored in the database in String form. For example, if you enter an email address, it is stored in the database as

```
"rfc822Name=email@yourCompany.com"
```

When the `subjectAltName` extension is added to the certificate, it is added in ASN.1 format. For example, a `subjectAltName` extension with only the email address above appears as follows:

```
SEQUENCE {  
    [1] "email@yourCompany.com",  
}
```

Security Manager Administration lets you view the `subjectAltName` component values in their String representation form (how they are stored in the database) and their ASN.1 decoded form (how they are added to the certificate).

You can also export the extension values in ASN.1 format in either binary or PEM-encoded format. This allows you to view the `subjectAltName` values using a binary editor to ensure that the information encoded in the certificate is as required by your application.

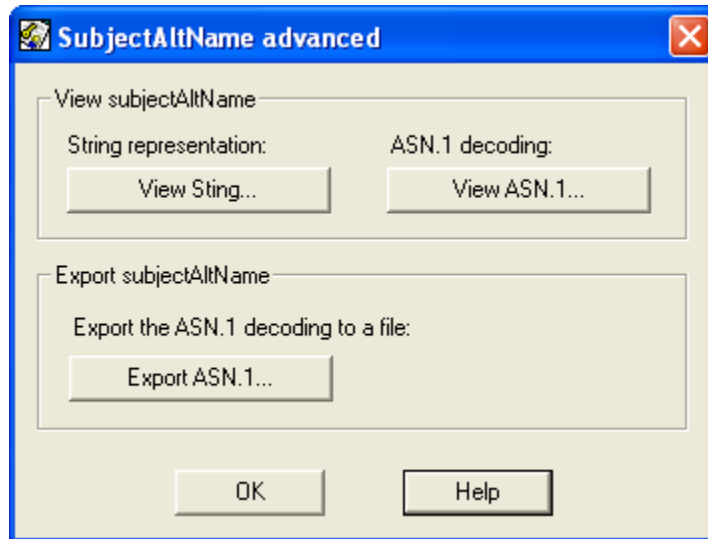
You can view and export `subjectAltName` component values in two ways:

- as stored with the user entry (["To view and export subjectAltName values in the user entry" on page 267](#))
- as stored in an existing user certificate (["To view and export subjectAltName values in the certificate" on page 268](#))

## To view and export subjectAltName values in the user entry

- 1 Log in to Security Manager Administration as an Entrust PKI administrator. See ["Logging in to Security Manager Administration" on page 46](#).
- 2 Find the user entry. For instructions, see ["Finding users" on page 134](#).
- 3 Right-click on the entry and click **Properties** to open the **User Properties** dialog box.
- 4 Click the **SubjectAltName** tab. Entries that you have added previously appear in the list.
- 5 Click **Advanced**.

The **SubjectAltName advanced** dialog box appears.

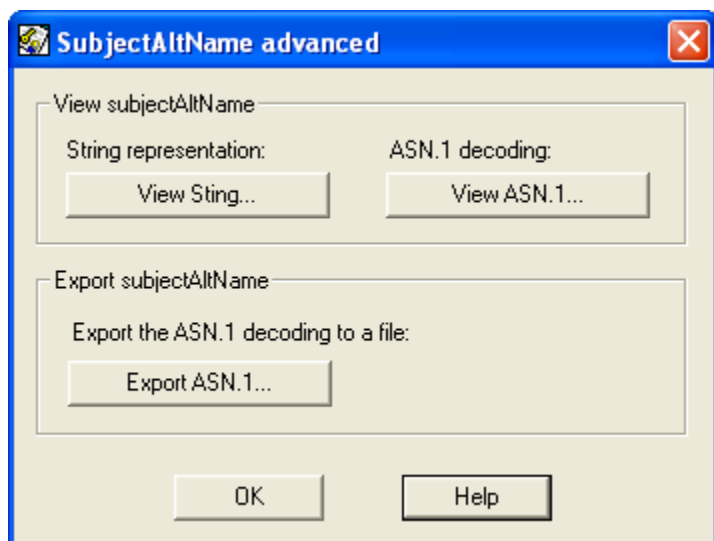


- 6 Use the appropriate button to view or export the `subjectAltName` values.
- 7 Click **OK**.

#### To view and export `subjectAltName` values in the certificate

- 1 Log in to Security Manager Administration as an Entrust PKI administrator. See [“Logging in to Security Manager Administration” on page 46](#).
- 2 Find the user entry. For instructions, see [“Finding users” on page 134](#).
- 3 Right-click on the entry and click **Properties** to open the **User Properties** dialog box.
- 4 Click the **Certificate List** tab.
- 5 Right-click the certificate and click **Certificate Contents**.
- 6 Click the **SubjectAltName** tab.
- 7 Click **Advanced**.

The **SubjectAltName advanced** dialog box appears.



- 8 Use the appropriate button to view or export the `subjectAltName` values.
- 9 Click **OK**.

# Excluding the subjectAltName from certificate definitions

By default, Security Manager adds the entire `subjectAltName` extension to any certificate created for a user. However, you may not want to have the entire `subjectAltName` or particular components in a certificate. For example, you may choose not to include the email address `subjectAltName` component in a signing certificate.

This section describes how you can edit the certificate definitions for a particular policy to exclude either the entire `subjectAltName` extension, or particular components.

## To exclude `subjectAltName` from a certificate definition

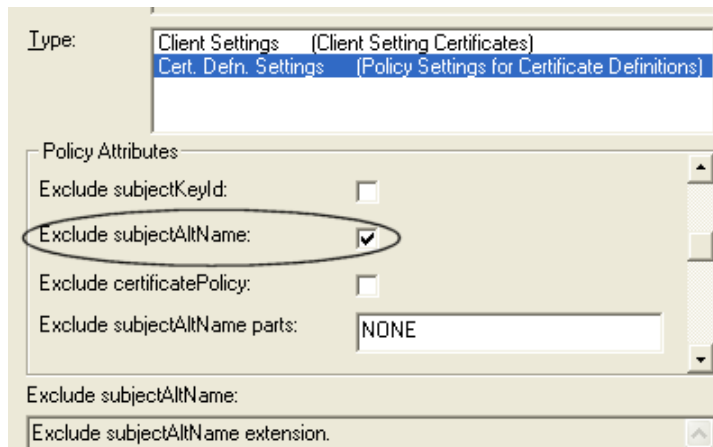
- 1 Log in to Security Manager Administration as an Entrust PKI administrator. See [“Logging in to Security Manager Administration” on page 46](#).
- 2 Go to **Security Policy > User Policies**.
- 3 Click on the user policy you want to modify.

---

**Note:** If you are modifying a default user policy, you may want to copy it first. To do so, right-click and select **Copy**.

---

- 4 To exclude the entire `subjectAltName` from the user policy, scroll down the **Policy Attributes** list and check **Exclude `subjectAltName`**.



The screenshot shows the 'Policy Attributes' section of the Security Manager Administration interface. The 'Type' dropdown is set to 'Cert. Defn. Settings (Policy Settings for Certificate Definitions)'. The 'Policy Attributes' section contains the following options:

- Exclude subjectKeyId: ☐
- Exclude subjectAltName: ☒ (This option is circled in the image)
- Exclude certificatePolicy: ☐
- Exclude subjectAltName parts: NONE (This is a dropdown menu)

Below the 'Policy Attributes' section, there is a section for 'Exclude subjectAltName:' with a checkbox for 'Exclude subjectAltName extension.' which is currently unchecked.

- 5 To exclude only certain `subjectAltName` components:
- a Scroll down the **Policy Attributes** list to find **Exclude subjectAltName parts**.

The screenshot shows a dialog box with a 'Type' dropdown menu at the top, currently showing 'Cert. Defn. Settings (Policy Settings for Certificate Definitions)'. Below this is a section titled 'Policy Attributes'. It contains four checkboxes: 'Exclude subjectKeyId:', 'Exclude subjectAltName:', 'Exclude certificatePolicy:', and 'Exclude subjectAltName parts:'. The 'Exclude subjectAltName parts:' checkbox is checked, and its corresponding text field contains the word 'NONE'. This text field is circled in red.

- b Delete **NONE** and enter one or more of the following, depending on which `subjectAltName` components you want to exclude:

- email
- dns
- dn
- uri
- ip
- oid
- upn
- other
- edi
- x400
- MsGUID
- FASC-N
- PermId

To exclude multiple components, separate each name with a space.

For information on these components, see [“SubjectAltName components” on page 251](#).

---

**Note:** If you enter `other`, all components of type `otherName` (for example, `MsGUID`, `upn`, `fascn`, `pid`) are excluded.

---

- 6 Click **Apply** and authorize your change.  
A success message appears.
- 7 Click **OK**.

Any new certificates created using this policy now exclude either the entire `subjectAltname` extension or the components you selected.



# Setting the criticality of the subjectAltName extension

By default, the `subjectAltName` extension is set up as non-critical. This means that when a client application validates a certificate, it does not have to process the information contained in the `subjectAltName` extension.

You can choose to set the `subjectAltName` extension as critical. For example, you may want to do so if your the subject line of the certificate is empty, or if you only want applications that can understand the `subjectAltName` to use the certificate.

Security Manager allows you to set criticality at a certificate definition policy level through Security Manager Administration.

## To set the criticality of the `subjectAltName` extension for a certificate definition

- 1 Log in to Security Manager Administration as an Entrust PKI administrator. See [“Logging in to Security Manager Administration” on page 46](#).
- 2 Go to **Security Policy > User Policies**.
- 3 Click on the user policy you want to modify.

---

**Note:** If you are modifying a default user policy, you may want to copy it first. To do so, right-click and select **Copy**.

---

- 4 To set the criticality of the `subjectAltName` for that user policy, scroll down the **Policy Attributes** list and check **SubjectAltName criticality**.

The screenshot shows the 'Policy Attributes' section of the Security Manager Administration interface. The 'Type' dropdown is set to 'Cert. Defn. Settings (Policy Settings for Certificate Definitions)'. In the 'Policy Attributes' list, the 'SubjectAltName criticality' checkbox is checked and circled. Other attributes include 'Exclude subjectAltName parts' set to 'NONE', 'Enable CMP override' (unchecked), and 'Allow unknown extensions' (unchecked). At the bottom, there is a text box for 'SubjectAltName criticality' with the instruction: 'Set the criticality of subjectAltName extension (1=critical 0=not critical(default)).'

- 5** Click **Apply** and authorize your change.  
A success message appears.
- 6** Click **OK**.

## Performing bulk operations

Bulk operations allow you to perform multiple add, modify, or delete actions in one operation. Only Entrust PKI administrators with the **Bulk and Report** permission (see [“Administering roles” on page 353](#)) can perform bulk operations.

Later sections of this chapter (beginning with [“Adding users in bulk to Security Manager” on page 288](#)) describe working examples of all the available bulk commands. For a list of bulk commands, see [“Bulk commands reference” on page 651](#).

This chapter contains the following sections:

- [“Bulk command syntax” on page 276](#)
- [“Creating bulk files” on page 279](#)
- [“Processing bulk files” on page 280](#)
- [“Viewing bulk output log files” on page 283](#)
- [“Advanced bulk processing” on page 285](#)
- [“Adding users in bulk to Security Manager” on page 288](#)
- [“Creating customized directory entries in bulk” on page 296](#)
- [“Setting up users for key recovery in bulk” on page 298](#)
- [“Deactivating, reactivating, and deleting users in bulk” on page 300](#)
- [“Revoking user certificates in bulk” on page 303](#)
- [“Changing user information in bulk” on page 308](#)
- [“Restoring information to the directory in bulk” on page 325](#)
- [“Updating users’ key pairs in bulk” on page 326](#)
- [“Notifying client applications in bulk” on page 327](#)
- [“Reissuing activation codes in bulk” on page 328](#)

# Bulk command syntax

Bulk operations are based on a scripting language called Tcl. You do not need to have worked with programming or scripting languages before to be able to run bulk commands using Tcl.

Later sections of this chapter (beginning with [“Adding users in bulk to Security Manager” on page 288](#)) describe working examples of all the available bulk commands. However, as with any scripting or programming language, learning some of the basic Tcl syntax rules can help you to understand how bulk commands work. For a list of bulk commands, see [“Bulk commands reference” on page 651](#).

This section contains the following topics:

- [“Commands and arguments” on page 276](#)
- [“Backslash” on page 276](#)
- [“Double quotation marks” on page 277](#)
- [“Curly braces” on page 277](#)
- [“Tcl output” on page 278](#)
- [“Additional resources about Tcl” on page 278](#)

## Commands and arguments

A Tcl command is formed by words separated by blank spaces:

```
command argument argument argument
```

The command indicates what action you are performing. The arguments provide additional information to a given command.

## Backslash

The backslash serves to keep commands and arguments together if they span more than one line. For example, the following statement

```
user ent_user "cn=Bob Jones + serialNumber=1QFB01, dc=Company \  
One, dc=com"
```

contains a backslash as the last character of the first line to indicate that the information on the second line belongs with the first line. The backslash must be the last character with no trailing spaces. If you do not include a backslash, then you are creating a line break.

You can use the backslash in other ways as well. One use is in quoting special characters. For example, the special character `\n` generates a new line. You can also use the backslash to turn off the special meanings of certain symbols, such as quotation marks, curly braces, square brackets, and dollar signs. Square brackets and

dollar signs are not covered in this section; see [“Additional resources about Tcl” on page 278](#).

## Double quotation marks

Use double quotation marks to group words into a single argument. For example, the following statement

```
user x "Alice Gray"
```

declares a user, Alice Gray. The double quotation marks indicate that the words “Alice” and “Gray” belong together as a single argument. Without the double quotation marks, the words “Alice” and “Gray” are read as two separate arguments.

Your argument can span multiple lines. If so, use the backslash (see [“Backslash” on page 276](#)). If you need to add the double quotation character into your argument, escape it with a backslash (for example, `\`”).

---

**Note:** It is important to use double quotation marks correctly. Failure to do so can cause errors.

---

## Curly braces

Use curly braces to group words into a single argument. The difference between double quotation marks and curly braces is that Tcl elements such as dollar signs and square brackets are interpreted inside double quotation marks, but are not interpreted inside curly braces. The importance of this is illustrated in the following bulk command statement:

```
user_import x {  
rp8$W79p9ShWPyhoWGpDmNbmheDapQT5yKx] 6Vn9xjXJfkIsJE03Jdhe43dg  
tfh594GcaktZdBS4Vg/Vju1Rt7TdlyjnMtlyojpP3XZ7+mCar6N6QGOJ8EC5o  
ivgj7o2wtzeGDHNMU+00x40/6lgnXITbnvr8Y0p0BBjCgXzRk9hCMRogeWAG  
imxs6LATVc03NfSaB5JcOX1jkEcE[615T2Id1WrOfAr6snmvHdaIugL9fv6xP  
a5AaMuhppBiIbrADJFQ41rXNn5qdT6Ck6QrCUBAJRL9wJUfWrxzmViRoliiV+  
pFBL/Rqrqy66V/lzhppfE5Xki6yurd8ukiaZk1AaX97z6yByMX1QjkDqiFvV6  
rtyKmJBS8rk/1P$K  
}
```

The text between the curly braces represents encrypted user information. The use of curly braces prevents any of the text inside the curly braces from being unintentionally interpreted as Tcl syntax.

The use of dollar signs and square brackets in Tcl is beyond the scope of this book. (See [“Additional resources about Tcl” on page 278](#).)

## Tcl output

The Bulk Console window shows the contents of the bulk log file, but this is not for standard output. Output from the Tcl `puts` command should be directed to a file.

## Additional resources about Tcl

The information provided in this “[Bulk command syntax](#)” section is enough to allow you to write simple scripts based on the examples provided in this chapter. However, if you want to write more advanced scripts, you need to know a bit more about Tcl and how it works. Here are some good places to start:

- *Practical Programming in Tcl and Tk Third Edition*, Brent B. Welsh, Prentice Hall PTR, ISBN 0-13-022028-0
- *Graphical Applications with Tcl & Tk Second Edition*, Eric Foster-Johnson, M&T Books, ISBN 1-55851-569-0
- <http://www.scriptics.com>
- <http://www.beedub.com/book>

# Creating bulk files

To perform bulk operations, you write scripts and save them in a bulk file. You then process the files in the Bulk Console window. To make writing bulk scripts easier, you can search for users and save the results to a text file. For information about finding users, see [“Finding users” on page 134](#).

The administrator processing the bulk file must have sufficient permissions to perform the operations specified in the bulk file. If your script contains a command that the administrator processing the bulk file does not have permission to perform, regardless of additional authorizations, the command is skipped and an error is logged in the bulk output log file.

Security Manager bulk operations use Tcl syntax. For more information about Tcl and bulk syntax, see [“Bulk command syntax” on page 276](#).

## To create a bulk file

- 1 Open a new file in any text editor.
- 2 Create a script by entering bulk commands.

For information about entering bulk commands in the bulk file, see [“Bulk command syntax” on page 276](#). Specific bulk commands are described in detail in [“Bulk commands reference” on page 651](#).

---

**Note:** Save the bulk command file as ASCII, not as Unicode. Security Manager Administration does not support Unicode.

---

- 3 Save the file with an `.entra` extension.

# Processing bulk files

After creating a bulk file (see [“Creating bulk files” on page 279](#)), you can process the bulk file in the Bulk Console. Only Entrust PKI administrators with the **Bulk and Report** permission can perform bulk operations (see [“Administering roles” on page 353](#)).

The administrator processing the bulk file must also have permission to perform the operations specified in the bulk file. If your script contains a command that you do not have permission to perform, regardless of additional authorizations, the command is skipped and an error is logged in the bulk output log file.

If you are performing a large bulk operation it is strongly recommended that you back up the Security Manager database before processing the bulk file. For information about backing up the database, see the *Security Manager Operations Guide*.

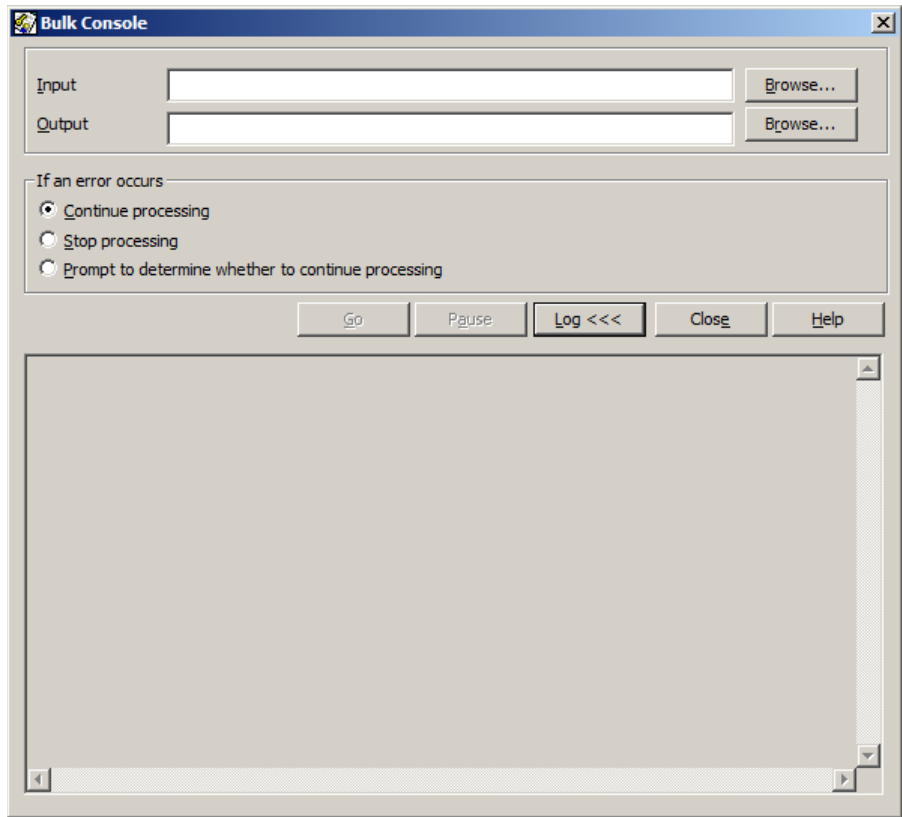
If an error occurs while processing, only the current operation fails. Operations successfully completed before the error occurred are saved and not overwritten.

## To process bulk files

- 1 Create a bulk file (see [“Creating bulk files” on page 279](#)).
- 2 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 3 Select **Operations > Bulk Console**.

The **Bulk Console** dialog box appears.





- 4 In the **Input script** field, enter the path and file name of your bulk file, or click Browse to locate the file.
- 5 In the **Output log** field, enter the path and file name of the bulk output log file or click Browse to choose a location and file name. Save the log file with a .log file extension.

For more information about the bulk output log file, see [“Viewing bulk output log files” on page 283](#).

- 6 In the **If an error occurs** pane:
  - If you want Security Manager Administration to continue processing when an error occurs, select **Continue processing**.
  - If you want Security Manager Administration to stop processing the bulk file when an error occurs, select **Stop processing**.
  - If you want Security Manager Administration to display a prompt when an error occurs, select **Prompt to determine whether to continue processing**.

The prompt displays the error and allows you to choose whether to continue processing the bulk file or to stop processing the bulk file.

- 7 To process the bulk file, click **Go**.

To halt the processing at any time, click **Pause**. Security Manager Administration halts processing as soon as it completes the current bulk command. To resume processing, click **Go**.

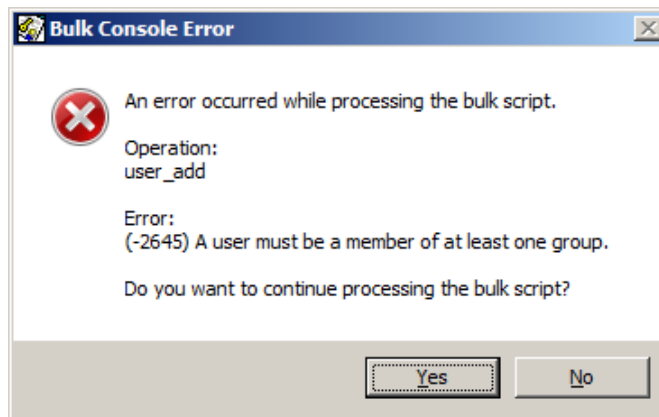
The Bulk Console window displays information about the progress of the bulk processing operation. To hide or display the information, click **Log**.

---

**Note:** Clicking **Close** before Security Manager Administration finishes processing a bulk file will halt processing and close the Bulk Console window (you are prompted to confirm the action). Closing the window before processing is complete will result in unfinished operations. Processing the bulk file again will result in errors. It is strongly recommended that you close the window only after you finish processing a bulk file.

---

- 8 If prompted, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).
- 9 If you selected the **Prompt to determine whether to continue processing** option and Security Manager encounters an error, the **Bulk Console Error** dialog box appears. For example:



To continue processing the bulk file, click **Yes**. To stop processing the bulk file, click **No**.

- 10 After Security Manager finishes processing the current bulk file, you can process another bulk file. Click **Reset** to reset all options to their defaults and repeat this procedure to process another bulk file.
- 11 To close the Bulk Console, clicking **Close**.

# Viewing bulk output log files

To view the results of processing a bulk command file, open the bulk output log (.log) file using a text editor. The log file is located in the directory you specified when you processed the bulk file (see [“Processing bulk files” on page 280](#)).

A success or failure message appears in the log file for every command you processed in the bulk file.

- [“Success messages” on page 283](#)
- [“Failure messages” on page 284](#)

## Success messages

The general syntax for a success message is:

```
[time-stamp] command-name:  
context-specific-information
```

Where:

- `time-stamp` is the time the command was executed in DD/MM/YYYY HH:MM:SS format.
- `command-name` is the bulk command that was processed.
- `context-specific-information` is information specific to each instance of a successfully processed bulk command. For example, each instance of a successfully processed `user_add` command includes the distinguished name (DN) and user activation codes of the newly added user.

---

**Note:** Some commands—such as `user_add`, `user_recover`, and `user_reissueactivationcodes`—return activation codes, which are saved to the bulk log file as part of the context specific information for the command. The activation codes only appear in the bulk log file if the Entrust PKI administrator running the bulk script has permission to view activation codes.

Because the activation codes are sensitive information, you must treat the bulk log file in a secure manner.

---

For example, the statement

```
[18/01/1999 12:06:27] user_add:  
cn=Alice Gray + serialNumber=1QBB01, dc=Company One, dc=com  
Reference Number: 45867568  
Authorization Code: JDU9-DS3S-SAWE
```

shows that user `cn=Alice Gray + serialNumber=1QBB01, dc=Company One, dc=com` was added at 12:06:27 on 18 January 1999, and was given reference number 45867568 and authorization code JDU9-DS3S-SAWE.

## Failure messages

The general syntax for a failure message is:

```
[time-stamp] command-name: ERROR
context-specific-error-message
```

Where:

- `time-stamp` is the time the command was processed in DD/MM/YYYY HH:MM:SS format
- `command-name` is the name of the bulk command
- `context-specific-error-message` is one or more lines of error message information specific to each instance of an unsuccessfully processed command

---

**Note:** Some commands, such as `user_add`, `user_recover`, and `user_reissueactivationcodes`, return activation codes, which are saved to the bulk log file as part of the context specific information for the command. The activation codes only appear in the bulk log file if the Entrust PKI administrator running the bulk script has permission to view activation codes.

Because the activation codes are sensitive information, you must treat the bulk log file in a secure manner.

---

For example, the statement

```
[18/01/1999 12:06:35] user_add: ERROR
cn=Bob Jones + serialNumber=1QFB01, dc=Company One, dc=com is not
found
```

indicates that the bulk operation to add user `cn=Bob Jones + serialNumber=1QFB01, dc=Company One, dc=com` failed at 12:06:35 on 18 January 1999 because the user's entry was not found in the Security Manager directory.

# Advanced bulk processing

Once you have become familiar with the bulk command syntax, you can start writing bulk operations in Tcl, a full-fledged scripting language. You can use Tcl to produce powerful, automated bulk scripts.

Included in the installation of Security Manager Administration are sample bulk scripts that illustrate ways of processing users in bulk using information from other sources, such as automatically generated reports. You can find these sample scripts in the `<SMA_Install_Dir>\bulk` folder.

One such example illustrates adding users using the information in a CSV (comma separated values) file. The CSV file contains each user's last name, first name, serial number, and email address. When you process the bulk script in the Bulk Console window, the script finds the CSV file and adds each user, based on the information in the CSV file.

An alternative method involves saving the results of a Find Users operation to a file through Security Manager Administration. For more information, see [“Finding users” on page 134](#).

If you intend to create users in the directory, but not to add them to Entrust (as you would when you move users from one CA to another), you can use the script shown here, as long as you do not include any of the commands that appear after the `user_createdirent` command. All the commands following the `user_createdirent` command are for adding the user.

## To add users to Security Manager using a CSV file

- 1 Using a database or spreadsheet application (such as Microsoft Excel), create a CSV file similar to the following:

```
# Line Format:
# Firstname, Lastname, serial number, email address
#
# Leading and trailing spaces will be trimmed from each value.
#
Alan, Aardvark, Alpha1, Alan.Aardvark@entrust.com
Bobbi, Bobcat, Bravo2, Bobbi.Bobcat@entrust.com
Cathy, Cuttlefish, Charlie3, Cathy.Cuttlefish@entrust.com
Douglas, Dog, Delta4, Douglas.Dog@entrust.com
Esther, Eagle, Echo5, Esther.Eagle@entrust.com
Frank, Fish, Foxtrot6, Frank.Fish@entrust.com
George, Gila Monster, Golf7, George.GilaMonster@entrust.com
Harry, Hippopotamus, Hotel8, Harry.Hippopotamus@entrust.com
```

Irene, Iguana, India9, Irene.Iguana@entrust.com  
 Jamie, Jackal, Juliet10, Jamie.Jackal@entrust.com  
 Katheryn, Kangaroo, Kilo11, Katheryn.Kangaroo@entrust.com  
 Larry, Llama, Lima12, Larry.Llama@entrust.com  
 Mary, Mongoose, Mike13, Mary.Mongoose@entrust.com  
 Niomi, Ngege, November14, Niomi.Ngege@entrust.com  
 Ophelia, Ostrich, Oscar15, Ophelia.Ostrich@entrust.com  
 Patti, Panda, Papa16, Patti.Panda@entrust.com  
 Quincy, Quagga, Quebec17, Quincy.Quagga@entrust.com  
 Randy, Racoon, Romeo18, Randy.Racoon@entrust.com  
 Sylvester, Snake, Sierra19, Sylvester.Snake@entrust.com  
 Tamara, Tiget, Tango20, Tamara.Tiget@entrust.com  
 Uretha, Ungulate, Uniform21, Uretha.Ungulate@entrust.com  
 Veronica, Viper, Victor22, Veronica.Viper@entrust.com  
 Wilma, Wolf, Whisky23, Wilma.Wolf@entrust.com  
 Xylia, Xtinct, X-Ray24, Xylia.Xtinct@entrust.com  
 Yolanda, Yak, Yankee25, Yolanda.Yak@entrust.com  
 Zeke, Zebra, Zulu26, Zeke.Zebra@entrust.com

**2** Save this file with the name `user_data.csv`.

**3** Create a bulk script similar to the following:

```
set fileid [open user_data.csv r]

while {[gets $fileid line] >=0} {

    if {[string compare [string index $line 0] "#"] == 0} {
        continue
    }

    set namedata [split $line ","]
    set firstName [string trim [lindex $namedata 0]]
    set lastName [string trim [lindex $namedata 1]]
    set serialNumber [string trim [lindex $namedata 2]]
    set mail [string trim [lindex $namedata 3]]

    user X
```

```

user_settemplate X Person

user_setattribute X cn + "$firstName $lastName"
user_setattribute X sn + $lastName
user_setattribute X serialNumber + $serialNumber
user_setattribute X mail + $mail

user_createdirententry X
user_setproperty X group + default
user_setproperty X role "End User"

user_add X

}

```

- 4** Modify this file according to your needs. For example, you may want to set key expiry dates as well, or add the users to specific groups as opposed to all groups.
- 5** Save the file with an `.entra` extension. The file can have any name (for example, `addusers.entra`). Ensure you save this file to the same location as the `user_data.csv` file.
- 6** Process the `.entra` file in the Bulk Console window. See [“Processing bulk files” on page 280](#).
- 7** Open the output log file to obtain the activation codes of each new user (see [“Viewing bulk output log files” on page 283](#)).
- 8** Turn to [“Managing activation codes” on page 153](#) for information on distributing a start-up package to each new user.

You have now added users to Security Manager using an advanced bulk script.

# Adding users in bulk to Security Manager

Adding users in bulk is a two-step process. First, you add the user through bulk processing. Then you distribute a start-up package to each user (see [“Managing activation codes” on page 153](#)).

---

**Note:** If you are using Security Manager with Microsoft Active Directory, then first add users to Active Directory using Microsoft tools. Once you add the users to Active Directory, you can find the users by directory attributes and add the users to Security Manager (see [“Adding existing users” on page 152](#)).

---

For background information about settings described in this section, see [“Creating new users” on page 146](#).

The following is an example of what your bulk script might look like:

```
user y
user_settemplate y Person
user_setattribute y cn + "Alice Gray"
user_setattribute y sn + Gray
user_setattribute y serialNumber + 1YCAG01
user_addtodn y serialNumber
user_setattribute y mail + alice.gray@company_one.com
user_setattribute y mail + alice.gray@anotheraddress.com
user_setparentdn y "dc=Company One Spinoff, dc=com"
user_setproperty y certificate_type Web web_default
user_setproperty y role "End User"
user_setproperty y group + "Group C"
user_setproperty y group + "Group G"
user_setproperty y key_expiry 31/03/2007 01/04/2008
user_setproperty y certificate_extension userStreet "100 East\
27th Street"
user_setproperty y certificate_extension userCity "New York"
user_setproperty y certificate_extension userEmail \
alice@company_one.com
user_createdirentry y
user_add y

user y
user_settemplate y Person
```



```
user_setattribute y cn + "Bob Jones"
user_setattribute y sn + Jones
...
#and so on
```

For information on using reports and scripts to generate the contents of the bulk command file, see [“Advanced bulk processing” on page 285](#).

---

**Note:** The bulk commands in the following procedure are presented as examples. Substitute the suggested variables and options according to your own requirements.

---

### To add users in bulk to Security Manager

- 1 Open a new file in any text editor.

- 2 Type the following:

```
user <userid>
```

Where <userid> is any unique combination of letters and numbers that you want. In the following steps you manipulate the user object with bulk commands in order to add a new user. For more information on user objects, see [“user” on page 659](#).

- 3 Set the user template type by typing

```
user_settemplate <userid> <usertypename>
```

For example:

```
user_settemplate <userid> Person
```

---

**Note:** The rest of this procedure assumes that you are adding users to the default Person user template. For more information, see [“Modifying the user template file and user types” on page 447](#).

---

- 4 Set the user’s common name (cn) attribute by typing

```
user_setattribute <userid> cn + "Alice Gray"
```

Substitute the name “Alice Gray” with the user’s first and last name. You must surround the first and last name by double quotation marks.

The cn attribute is a required attribute, based on the settings in [“Creating new users” on page 146](#).

- 5 Set the user’s surname (sn) attribute by typing

```
user_setattribute <userid> sn + Gray
```

Substitute the name `Gray` with the user's last name. No quotation marks are required unless the user's last name is made up of two or more words.

The `sn` attribute is a required attribute, based on the settings in ["Creating new users" on page 146](#).

**6** Optionally, set the user's serial number attribute by typing

```
user_setattribute <userid> serialNumber + 1YCAG01
```

Substitute `1YCAG01` with the user's serial number. The serial number can be any value that is used to uniquely identify the user, such as an employee number.

**7** Optionally, you can add the serial number to the user's DN by typing the following:

```
user_addtodn <userid> serialNumber
```

This command makes the user's serial number appear in the user's DN. According to the default settings for adding a new user, the serial number is the only attribute that you can add to the user's DN. To change the default settings, see ["Modifying the user template file and user types" on page 447](#).

---

**Note:** The default user template when using Active Directory Application Mode (ADAM) or Active Directory Lightweight Directory Services (AD LDS) does not allow you to add the serial number to the user's DN. ADAM and AD LDS do not support multi-valued RDNs.

---

The `user_addtodn` command is optional.

**8** Optionally, set the user's email attribute by typing

```
user_setattribute <userid> mail + alice.gray@company_one.com
```

```
user_setattribute <userid> mail + alice.gray@anotheraddress.com
```

Substitute `alice.gray@company_one.com` and `alice.gray@anotheraddress.com` with the user's email address(es).

---

**Note:** This example shows the addition of two email addresses for Alice Gray. You can set zero, one, or multiple email addresses for your users.

---

**9** Optionally, you can specify the DN under which the user is created by typing

```
user_setparentdn <userid> "dc=Company One Spinoff, dc=com"
```

Substitute `"dc=Company One Spinoff, dc=com"` with the DN under which you want to create the user. You must surround the DN with double quotation marks.

By default, users are created under the DN of the CA (for example, `cn=Alice Gray, dc=Company One, dc=com`). The `user_setparentdn` command allows you to create a new user under the DN of your choice (for example, `cn=Alice Gray, dc=Company One Spinoff, dc=com`).

## 10 Optionally, set the user's certificate category by typing

```
user_setproperty <userid> certificate_type <certcategory>
<certtype>
```

For example, to specify the Web certificate category and the `web_default` certificate type, type:

```
user_setproperty <userid> certificate_type Web web_default
```

This example bulk command statement assumes that you are using Web licenses.

If you do not set the certificate category and type, by default the user is assigned the Enterprise certificate category, and the `ent_default` certificate type. For more information on certificate categories and types, see ["Customizing certificates" on page 525](#).

## 11 Set the user's role by typing

```
user_setproperty <userid> role <roleUniqueName>
```

For example:

```
user_setproperty <userid> role "End User"
```

You must surround the term `End User` or any other role name of more than one word, with double quotation marks.

## 12 Designate which group the user belongs to by typing

```
user_setproperty <userid> group + default
```

This statement makes the user a member of the default group.

If you would like the user to belong to all groups that currently exist and any future groups, use an asterisk to represent all current and all future group names, as follows:

```
user_setproperty <userid> group + *
```

If you would like the user to belong only to certain specific groups, you must include a bulk command statement for each group the user is to belong to. For example, you can add the user to Group C and Group G as follows (double quotation marks must surround any group names of more than one word):

```
user_setproperty <userid> group + "Group C"
```

```
user_setproperty <userid> group + "Group G"
```

For more information on groups, see ["Administering groups" on page 329](#).

## 13 Optionally, add the `subjectAltName` extension values to the entry by typing

```
user_setproperty <userid> alternate_id +
"<subjectAltNameComponent>=<Value>"
```

where `<subjectAltNameComponent>=<Value>` is the String representation of the `subjectAltName` component name and value. To determine the String representation, complete the following steps:

- a In Security Manager Administration, click **Operations > Create subjectAltName**.

The **Create subjectAltName** dialog box opens.

- b Add the `subjectAltName` component values, as described in [Table 17 on page 251](#).

---

**Note:** When adding values, Security Manager automatically escapes special characters such as > or " using the backslash (\) when you click **OK**. If your string includes a backslash, you must prefix it with another backslash when you enter it.

---

---

**Note:** Security Manager Administration does not check to ensure the value is correct. If component values are incorrect, the certificates created for the user may fail authentication. Should this occur, you must edit the `subjectAltName` component and reissue the keys and certificates.

---

- c Once you have added all component values, click the **View String** button.
- d Copy the entry into the command line.

For example:

```
user_setproperty <userid> alternate_id +  
"rfc822Name=alice.grey@companyone.com" "userPrincipalName=agrey"  
"rfc822Name=alice.grey@home.com"
```

---

**Note:** If you have set up auto-population from a directory ("[Configuring auto-population of the subjectAltName from the directory](#)" on page 257) and modified the user template file to make directory attributes that are mapped to the `subjectAltName` mandatory, you may not need to use this command when creating users. The auto-population feature ensures that the `subjectAltName` extension is created.

If you do use this command to add more values to the `subjectAltName` than are stored in the directory, ensure that the `<only_if_absent>` parameter is set correctly ([Step 3 on page 258](#)). For example, if you have kept the default email mapping, `<only_if_absent>` is set to 1, and you have the following commands in your script: `user_setattribute x mail + alice.grey@companyone.com` `user_setproperty x alternate_id + email=alice.grey@home.com`. The mail address (alice.grey@companyone.com) is not added to `subjectAltName`. If `<only_if_absent>` is set to 0, it is added to `subjectAltName`.

---

## 14 Optionally, set the user's key lifetimes by typing

```
user_setproperty <userid> key_lifetimes 24 30 50
```

The values 24, 30, and 50 represent the encryption public key lifetime, the verification public key lifetime, and the signing private key lifetime respectively. Set these values according to your requirements. The encryption and verification public key lifetimes are set in terms of months, with a possible range of 2 to 60 months. The signing private key lifetime is set as a percentage of the verification public key lifetime, with a possible range of 1 to 100 percent.

Key lifetimes and key expiry dates are mutually exclusive. Setting key lifetimes overwrites any previously established key expiry dates. (The opposite is also true.) If you prefer to set key expiry dates, skip this step and proceed to [Step 15 on page 293](#).

If you do not set the user's key lifetimes, the user receives the default settings, as described in ["Configuring options for the various certificate categories" on page 97](#).

### 15 Optionally, set the key expiry date by typing

```
user_setproperty <userid> key_expiry 03/28/2007 04/17/2007
```

The value 03/28/2007 represents the expiry date of both the user's encryption public key and the user's signing private key. The value 04/17/2001 represents the expiry date of the user's verification public key. Set these values according to your requirements. The format for the date is dependent on regional settings. For example, the default for English (US) is MM/DD/YYYY (for example, 12/25/2007).

Key expiry dates and key lifetimes are mutually exclusive. Setting key expiry dates overwrites any previously established key lifetimes. (The opposite is also true.) If you prefer to set key lifetimes, return to [Step 14 on page 292](#).

If you do not set the user's key expiry dates, the user receives the default key lifetime settings, as described in ["Configuring options for the various certificate categories" on page 97](#).

### 16 Optionally, set certificate extension variables by typing

```
user_setproperty <userid> certificate_extension <varID> <varvalue>
```

To specify a certificate variable, you must use the variable's Var ID, as defined in the master.certspec file. For more information on certificate variables and the master.certspec file, see ["Customizing certificates" on page 525](#).

For example, to set the certificate variable userStreet and value 100 East 27th Street:

```
user_setproperty y certificate_extension userStreet "100 East\
27th Street"
```

Another example:

```
user_setproperty y certificate_extension userEmail \
alice@company_one.com
```

In this command statement, `alice@company_one.com` is the variable assigned to the certificate variable `userEmail`. No double quotation marks are required because `alice@company_one.com` appears as a single unbroken word.

---

**Note:** When a certificate variable requires multiple values, always specify the values as a space-separated list enclosed in double quotation marks.

---

**17** Create the user's directory entry by typing

```
user_createdirectory <userid>
```

**18** Add the user by typing

```
user_add <userid>
```

**19** Repeat [Step 2](#) to [Step 18](#) for each user you want to add.

**20** Save the file with an `.entra` extension. The file can have any name. Your file should appear as follows:

```
user x
user_settemplate <userid> Person
user_setattribute <userid> cn + "Alice Gray"
user_setattribute <userid> sn + Gray
user_setattribute <userid> serialNumber + 1YCAG01
user_setattribute <userid> mail + alice.gray@company_one.com
user_setproperty <userid> role "End User"
user_setproperty <userid> group + default
user_createdirectory <userid>
user_add <userid>

user <userid>
user_settemplate <userid> Person
user_setattribute <userid> cn + "Bob Jones"
user_setattribute <userid> sn + Jones
...
#and so on
```

**21** Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).

**22** Open the output log file to obtain the activation codes of each new user (see [“Viewing bulk output log files” on page 283](#)).

---

**Note:** The activation codes only appear in the bulk log file if the Entrust PKI administrator running the bulk script has permission to view activation codes. Because the activation codes are sensitive information, you must treat the bulk log file in a secure manner.

---

- 23** Turn to [“Managing activation codes” on page 153](#) for information on distributing a start-up package to each new user.
- You have now added users with default settings using bulk commands.

# Creating customized directory entries in bulk

An Entrust PKI administrator can add customized user entries to the directory through bulk processing. A customized user entry is an entry in the directory that is not based on a predefined user type such as Person or Web Server. To add a customized directory entry, you must

- specify the complete distinguished name of the user in the `user` command, and also include the `New` argument
- specify values for the `objectClass` attribute, instead of using the `user_settemplate` command

If you do not add all the values to the `objectClass` attribute that the directory schema requires, the `user_createdirententry` command may fail.

- specify values for any other required attributes
- use the `user_createdirententry` command to create the entry in the directory

You must be careful when creating customized directory entries. If you create entries using an uncommon object class, you might encounter difficulties when searching for those entries. Also, when you change the distinguished name of an entry, Security Manager tries to match the entry to a template. Errors may occur if the entry does not match any of your templates.

## To create customized Directory entries

- 1 Open a new file in any text editor.
- 2 Declare the DN of the new directory entry by typing the following, for example:

```
user z "cn=Bob Jones + serialNumber=1QFB01, dc=Company \
One, dc=com" New
```

Substitute the DN in this statement with the DN of the entry you want to create. Include the argument `New` after the DN. Remember to surround the user's DN with double quotation marks. The backslash keeps the information in the first and second lines together as a single bulk command statement.

- 3 Add values to the `objectClass` attribute by typing

```
user_setattribute z objectClass + organizationalPerson
user_setattribute z objectClass + Person
```

Substitute the values `organizationalPerson` and `Person` with values you want to add to the `objectClass` attribute. Also, add any other values to the `objectClass` attribute that your directory schema requires.

---

**Note:** If you do not add all the values to the `objectClass` attribute that the directory schema requires, the `user_createdirententry` command may fail.

---



- 4 Set attributes for the user by typing the following, for example:

```
user_setattribute z sn + Jones
user_setattribute z mail + jones@company_one.com
```

Substitute these attributes and add any others according to your requirements for this directory entry. Attributes are set in the same way as any regular directory entry. For other examples, see [“Adding users in bulk to Security Manager” on page 288](#).

- 5 Create the directory entry by typing

```
user_createdirentory z
```

- 6 If you wish to add the entry, type the following, for example:

```
user_setproperty z role "End User"
user_setproperty z group + default
user_add z
```

Substitute the values for the role and group as required. You can set any other properties here as well. For more information on setting properties, see [“Changing user properties in bulk” on page 314](#).

If you want to create a directory entry but do not want to add the entry, do not include the commands in this step.

- 7 Repeat [Step 2](#) to [Step 6](#) for all the customized directory entries you want to create.
- 8 Save the file with an `.entra` extension. The file can have any name (for example, `customentries.entra`).
- 9 Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).

You have now created customized directory entries in bulk.

# Setting up users for key recovery in bulk

You must set up a user for key recovery under the following circumstances:

- When a user forgets a password. This is the most common occurrence.
- When an Entrust profile is lost or damaged.
- When a user believes that keys are compromised or that attackers possess passwords or Entrust profiles.
- When a user is set up not to have key pairs automatically updated and the user's situation changes. For example, when a contractor's contract is extended and you need to issue new keys for the extension period.
- When a user's signing private key expires (which should rarely or never occur).

For more information, see [“Recovering user key pairs” on page 162](#).

Topics in this section:

- [“Creating the bulk file and performing key recovery” on page 298](#)
- [“Canceling key recovery in bulk” on page 299](#)

## Creating the bulk file and performing key recovery

Complete the following procedure to create the bulk file and perform key recovery on those users.

### To set up users for key recovery in bulk

- 1** Open a new file in any text editor.
- 2** Identify the user you want to recover by typing the following, for example:  

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user you want to recover. Remember to surround the user's DN with double quotation marks.
- 3** Recover the user by typing  

```
user_recover z
```
- 4** Repeat [Step 2](#) and [Step 3](#) for all the users you want to recover.
- 5** Save the file with an `.entra` extension. The file can have any name (for example, `keyrecover.entra`).
- 6** Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).
- 7** Open the output log file to obtain the activation codes of each recovered user (see [“Viewing bulk output log files” on page 283](#)).

---

**Note:** The activation codes only appear in the bulk log file if the Entrust PKI administrator running the bulk script has permission to view activation codes. Because the activation codes are sensitive information, you must treat the bulk log file in a secure manner.

---

- 8** Distribute the activation codes to each recovered user.  
You have now set up users for key recovery in bulk.

## Canceling key recovery in bulk

The ability to cancel key recovery is very useful when users

- manage to restore their Entrust profiles from backup copies
- remember their passwords

In either case, there is no need to continue the key recovery process. Canceling key recovery returns users to the Active state.

### To cancel key recovery for users in bulk

- 1** Open a new file in any text editor.
- 2** Identify the user for whom you want to cancel key recovery by typing the following, for example:  

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user for whom you want to cancel key recovery. Remember to surround the user's DN with double quotation marks.
- 3** Cancel key recovery for the user by typing  

```
user_cancelrecover z
```
- 4** Repeat [Step 2](#) and [Step 3](#) for all the users for whom you want to cancel key recovery.
- 5** Save the file with an `.entra` extension. The file can have any name (for example, `cancelkeyrecover.entra`).
- 6** Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).  
You have now canceled key recovery for users in bulk.

# Deactivating, reactivating, and deleting users in bulk

Deactivating, revoking, and deleting users and entities are related tasks that restrict users and entities from using Security Manager—or that remove them from the system altogether. You can restrict users and entities from using Entrust in several ways:

- you can deactivate them
- you can deactivate them and revoke their certificates
- you can deactivate them, revoke their certificates, and delete their Directory entry
- you can suspend their certificates

For more detailed information, see [“Deactivating and reactivating users” on page 172](#).

Topics in this section:

- [“Deactivating users in bulk” on page 300](#)
- [“Reactivating users in bulk” on page 301](#)
- [“Deleting users from the directory in bulk” on page 301](#)

## Deactivating users in bulk

For general information about deactivating users, see [“Deactivating users” on page 172](#).

### To deactivate users in bulk

- 1 Open a new file in any text editor.
- 2 Identify the user you want to deactivate by typing the following, for example:  

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user you want to deactivate. Remember to surround the user's DN with double quotation marks.
- 3 Deactivate the user by typing  

```
user_deactivate z
```
- 4 Repeat [Step 2](#) and [Step 3](#) for all the users you want to deactivate.
- 5 Save the file with an `.entra` extension. The file can have any name (for example, `deactivate.entra`).
- 6 Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).

You have now deactivated users in bulk.

## Reactivating users in bulk

For general information about reactivating users, see [“Reactivating users” on page 173](#).

### To reactivate users in bulk

- 1 Open a new file in any text editor.
  - 2 Identify the user you want to reactivate by typing the following, for example:  

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user you want to reactivate. Remember to surround the user's DN with double quotation marks.
  - 3 Reactivate the user by typing  

```
user_reactivate z
```
  - 4 Repeat [Step 2](#) and [Step 3](#) for all the users you want to reactivate.
  - 5 Save the file with an `.entra` extension. The file can have any name (for example, `reactivate.entra`).
  - 6 Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).
- You have now reactivated users in bulk.

## Deleting users from the directory in bulk

You must deactivate users before you can delete them from the directory. Deleting a user removes the user's entry from the directory.

---

**Note:** If you are using Security Manager with Microsoft Active Directory, use your Microsoft Active Directory tools to delete user entries from the directory.

---

### To delete users in bulk

- 1 Open a new file in any text editor.
- 2 Identify the user you want to delete by typing the following, for example:  

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user you want to delete. Remember to surround the user's DN with double quotation marks.
- 3 Delete the user by typing

```
user_deletedirentry z
```

- 4** Repeat [Step 2](#) and [Step 3](#) for all the users you want to delete.
- 5** Save the file with an `.entra` extension. The file can have any name (for example, `deleteusers.entra`).
- 6** Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).

You have now deleted users from the directory in bulk.

# Revoking user certificates in bulk

Bulk operations allow you to

- revoke user certificates in bulk
- put user certificates on hold in bulk
- take user certificates off hold in bulk

Topics in this section:

- [“Revoking user certificates in bulk” on page 303](#)
- [“Putting user certificates on hold in bulk” on page 305](#)
- [“Taking certificates off hold in bulk” on page 306](#)

## Revoking user certificates in bulk

For general information on revoking user certificates, see [“Revoking user certificates” on page 174](#).

---

**Note:** The bulk commands in the following procedure are presented as examples. Substitute the suggested variables and options according to your own requirements.

---

### To revoke user certificates in bulk

- 1 Open a new file in any text editor.
- 2 Identify the user whose certificates you want to revoke by typing, for example:  

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user whose certificates you want to revoke. Remember to surround the user's DN with double quotation marks.
- 3 If you want to revoke the user's certificates due to a key compromise, continue with [Step 4](#). If you want to revoke the user's certificates for any other reason, skip to [Step 5](#).
- 4 Revoke the user's certificates due to a key compromise by typing  

```
user_revoke z "Entrust password was compromised" KC All Both \
11/04/2001
```

  - Substitute the comment “Entrust password was compromised” with a comment appropriate to the situation. If you do not include a comment, the comment is set by default to `unspecified`.
  - `KC` indicates the keys are being revoked due to key compromise.

- `All` indicates that all keys are being revoked.  
If you want to revoke only the latest keys, use the term `Latest` instead of the term `All`. If you do not specify a term, the default is `All`.
- `Both` indicates that both encryption and verification keys are being revoked.  
If you want to revoke only the encryption keys, use the term `E` instead of the term `Both`. If you want to revoke only the verification keys, use the term `V` instead of the term `Both`. If you do not specify a term, the default is `Both`.
- The date indicates the last day the keys were known to be uncompromised.  
If you do not specify a date, the issuing date in the certificate is used.
- The backslash at the end of the first line keeps the information on the first and second lines together as a single bulk command statement.

Now skip to [Step 6](#).

## 5 Revoke the user's certificates for any other reason by typing

```
user_revoke z "Contract ended" COO All Both
```

Where:

- Substitute the comment `Contract ended` with a comment appropriate to the situation. If you do not include a comment, the comment is set by default to `unspecified`.
- The term `COO` indicates the keys are being revoked due to a Cessation of Operation. The other possible terms are:
  - `AC` (affiliation change)
  - `S` (superseded)
  - `OH` (on hold)
  - `U` (unspecified).

If you do not include a reason, the reason for revoking is set by default to `U` (unspecified).

- `All` indicates that all keys are being revoked.  
If you want to revoke only the latest keys, use the term `Latest` instead of the term `All`. If you do not specify a term, the default is `All`.
- `Both` indicates that both encryption and verification keys are being revoked.
  - If you want to revoke only the encryption keys, use the term `E`.
  - If you want to revoke only the verification keys, use the term `V`.
 If you do not specify a term, the default is `Both`.

## 6 Repeat [Step 2](#) to [Step 5](#) for all the user certificates you want to revoke.

## 7 Save the file with an `.entra` extension. The file can have any name (for example, `certrevoke.entra`).



- 8 Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).
- 9 To issue a new certificate revocation list, see [“Issuing a new CRL after revoking a certificate” on page 180](#).

You have now revoked user certificates in bulk.

## Putting user certificates on hold in bulk

Security Manager lets you permanently revoke certificates when they are no longer trusted, but you can also suspend certificates (put them on hold). For example, if a user misplaces a smart card, you can suspend the certificates on that card. If the user finds the card, you can take the certificates off hold. If the user does not find the smart card, you can permanently revoke the certificates.

When you suspend a certificate, Security Manager adds an entry to the CRL for the certificate. The entry indicates that the certificate is suspended, and is not trusted at this time.

---

**Note:** The effects of suspending a certificate are significant. A suspended certificate is not trusted in the same way that revoked certificates are not trusted.

---

For information on suspending certificates one at a time, see [“Suspending user certificates” on page 181](#).

### To put user certificates on hold in bulk

- 1 Open a new file in any text editor.
- 2 Identify the user whose certificates you want to suspend by typing, for example:

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user whose certificates you want to suspend. Remember to surround the user's DN with double quotation marks.

- 3 Suspend the user's certificates by typing

```
user_revoke z "Misplaced smartcard" OH All Both
```

- Substitute the comment `Misplaced smartcard` with a comment appropriate to the situation.

If the comment contains a space, you must enclose it in quotation marks. If you do not include a comment, the comment is set by default to `unspecified`.

- `OH` indicates the keys are being placed on hold.
- `All` indicates that all keys are being revoked.

If you want to revoke only the latest keys, use the term `Latest` instead of the term `All`. If you do not specify a term, the default is `All`.

- `Both` indicates that both encryption and verification keys are being revoked.

If you want to revoke only the encryption keys, use the term `E` instead of the term `Both`. If you want to revoke only the verification keys, use the term `V` instead of the term `Both`. If you do not specify a term, the default is `Both`.

To omit an argument, use empty quotes. The following example omits the comment argument:

```
user_revoke z "" OH All Both
```

- 4 Repeat [Step 2](#) and [Step 3](#) for all the user certificates you want to suspend.
- 5 Save the file with an `.entra` extension. The file can have any name (for example, `cert_hold.entra`).
- 6 Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).
- 7 To issue a new certificate revocation list, see [“Issuing a new CRL after revoking a certificate” on page 180](#).

You have now put user certificates on hold using bulk commands. The certificates are no longer trusted.

## Taking certificates off hold in bulk

You can resolve certificate suspensions by taking them off hold.

### To take user certificates off hold in bulk

- 1 Open a new file in any text editor.
- 2 Identify the user whose certificates you want to take off hold by typing, for example:

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user whose certificates you want to take off hold. Remember to surround the user's DN with double quotation marks.

- 3 Take the user's certificates off hold by typing

```
user_cancelhold z 994090853 E Y
```

- Substitute the serial number of the certificate in the example with the serial number of the certificate you want to take off hold.
- The term `E` indicates that the encryption certificate is being taken off hold. If you want to revoke only the verification certificate, use the term `V` instead of the term `E`.

- The term `Y` indicates to issue the CRL immediately. If you do not want immediate issue of the CRL, use the term `N` instead of the term `Y`.

---

**Note:** You must specify a value for every argument of the `user_cancelhold` command.

---

- 4 Repeat [Step 2](#) and [Step 3](#) for all the user certificates you want to suspend.
- 5 Save the file with an `.entra` extension. The file can have any name (for example, `cert_offhold.entra`).
- 6 Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).

You have now taken user’s certificates off hold. The certificates are trusted.

# Changing user information in bulk

This section describes bulk commands you can use to change user information. You can

- change users' DNs (see [“Changing users' DNs in bulk” on page 308](#))
- [“Canceling the change DN operation in bulk” on page 313](#)
- change the properties on user certificates (see [“Changing user properties in bulk” on page 314](#))
- add and delete users' Directory attributes (see [“Adding and deleting directory attributes in bulk” on page 323](#))

## Changing users' DNs in bulk

You can change a user's DN in bulk using two different methods. The first method involves creating a new directory entry. The user is then assigned the DN of the new directory entry. This method leaves the old entry in the directory. You may want to keep the old directory entry for other administrative purposes, or you may want to delete it. The second method of changing a user's DN involves renaming the existing directory entry. The second method is much easier, but is only possible if you are using LDAPv3 to communicate between the directory and Security Manager. (Whether you are using LDAPv2 or LDAPv3 was determined during installation.)

To use the first method, see [“To change users' DNs in bulk by creating new directory entries” on page 309](#). To use the second method, see [“To change users' DNs in bulk by renaming the directory entries” on page 310](#).

When renaming a DN, there are often other directory attributes that are not part of the DN, but that you must also change. For example, if a person's last name changes, you must change both the common name (cn) and the surname (sn) attributes (that is, assuming the user has an sn attribute). If the cn attribute is part of the user's DN, the Change DN operation updates the cn attribute. However, you must add a separate `user_setattribute` command to update the sn attribute. This applies specifically to renaming Directory entries, and is illustrated in [“To change users' DNs in bulk by renaming the directory entries” on page 310](#).

---

**Note:** If you are using Security Manager with Microsoft Active Directory, then you change the DN in Active Directory using Microsoft tools, and then assign the new DN to the user in bulk (see [“To assign new DNs to users in bulk” on page 312](#)).

---

For more general information on changing users' DNs, including what user states allow a DN change, see [“Changing distinguished names” on page 193](#).

You can also cancel the Change DN operation. For more information, see [“Canceling the change DN operation in bulk” on page 313](#).

The following procedures show sample bulk commands. Substitute the suggested variables and options according to your own requirements.

### To change users' DNs in bulk by creating new directory entries

- 1 Open a new file in any text editor.

- 2 Type the following:

```
user x
```

This statement declares a user object with a `userID` of `x`. You can set the `userID` to any combination of letters and numbers that you want. In the following steps you manipulate user object `x` with bulk commands in order to create a new Directory entry.

- 3 Set the user template type to `Person` by typing

```
user_settemplate x Person
```

- 4 Set the user's common name (`cn`) attribute by typing

```
user_setattribute x cn + "Alice Jones"
```

Substitute the name "Alice Jones" with the user's first and last name. If you are changing the user's DN due to a name change, use the new name. You must surround the first and last name with double quotation marks.

- 5 Set the user's surname (`sn`) attribute by typing

```
user_setattribute x sn + Jones
```

Substitute the name `Jones` with the user's last name. If you are changing the user's DN due to a changed last name, set the `sn` attribute with the user's new last name. You do not need to include quotation marks unless the user's last name is made up of two or more words.

- 6 Set the user's serial number attribute by typing

```
user_setattribute x serialNumber + AJYC123
```

Substitute `AJYC123` with the user's serial number. If the user's serial number has changed, set the serial number attribute with the new serial number.

You can set the serial number to any value that is used to uniquely identify the user, such as an employee number. By default, this attribute is optional.

- 7 Optionally, set the user's email attribute by typing

```
user_setattribute x mail + alice.jones@company_one.com
```

Substitute `alice.jones@company_one.com` with the user's email address. If the user's email address has changed, set the email attribute with the new email address.

- 8 Set the user's parent DN by typing the following:

```
user_setparentdn x "ou=Product Management,dc=Company One,dc=com"
```

Substitute "ou=Product Management, dc=Company One, dc=com" with the user's parent DN. The parent DN is any part of the DN that does not include the common name (that is, l=(locality), ou=(organizational unit), o=(organization), c=(country), and so on). Remember to surround the user's parent DN with double quotation marks.

If part of the user's parent DN has changed (for example, the user has moved from ou=Engineering to ou=Product Management), set the parent DN to reflect the latest changes.

If the user's parent DN has not changed, you can skip this step.

**9** Create the new entry in the directory by typing

```
user_createdirentry x
```

**10** Identify the user whose DN you want to change by typing

```
user x "cn=Alice Gray,ou=Engineering,dc=Company One,dc=com"
```

Substitute "cn=Alice Gray, ou=Engineering, dc=Company One, dc=com" with the user's old DN (the DN that you are changing). Remember to surround the user's DN with double quotation marks.

**11** Assign the new DN to the user by typing

```
user_assigndn x "cn=Alice Jones,ou=Product Management,dc=Company One,dc=com"
```

Substitute "cn=Alice Jones, ou=Product Management, dc=Company One,dc=com" with the user's new DN. Remember to surround the user's DN with double quotation marks. The backslash that appears at the end of the first line keeps the information on the first and second lines together.

**12** Repeat [Step 2](#) to [Step 11](#) for all the users whose DNs you want to change.

**13** Save the file with an .entra extension. The file can have any name (for example, changedn.entra).

**14** Process this file in the Bulk Console window. See ["Processing bulk files" on page 280](#).

The user must log in to an Entrust desktop application to complete the change DN operation.

You have now changed users' DNs in bulk by creating a new directory entry.

### To change users' DNs in bulk by renaming the directory entries

**1** Open a new file in any text editor.

**2** Identify the user whose DN you want to change by typing

```
user z "cn=Alice Gray,ou=Engineering,dc=Company One,dc=com"
```

Substitute the DN in this statement with the current DN of the user whose DN you want to change. Remember to surround the user's DN with double quotation marks.

**3** Update any attributes that do not appear in the DN by typing

```
user_setattribute z sn + Jones
```

```
user_setattribute z sn - Gray
```

The first line in this step adds the value `Jones` to the surname (`sn`) attribute. The second line removes the value `Gray`. You must update any other attributes that do not appear in the DN, but that require updating due to the Change DN operation in the manner illustrated here. (For example, if the user's name has changed, their email address attribute may need updating.) Any attributes that appear in the DN (for example, the common name (`cn`) attribute or the serial number) are updated in the following steps.

**4** Rename the user's directory entry by typing

```
user_renamedirectory z "cn=Alice Jones,ou=Product Management, \
dc=Company One,dc=com"
```

Substitute "`cn=Alice Jones, ou=Product Management, dc=Company One, dc=com`" with the user's new DN. Remember to surround the user's DN with double quotation marks.

In this example, the new DN reflects a change in the user's common name (from Alice Gray to Alice Jones), and a change for the user within the organization (from Engineering to Product Management).

The backslash at the end of the first line keeps the information on the first and second lines together.

**5** Assign the new DN to the user by typing

```
user_assigndn z "cn=Alice Jones,ou=Product Management,dc=Company
One,dc=com"
```

Substitute "`cn=Alice Jones, ou=Product Management, dc=Company One, dc=com`" with the user's new DN. Remember to surround the user's DN with double quotation marks.

The backslash at the end of the first line keeps the information on the first and second lines together.

**6** Repeat [Step 2](#) to [Step 5](#) for all the users whose DNs you want to change.

**7** Save the file with an `.entra` extension. The file can have any name (for example, `changedn.entra`).

**8** Process this file in the Bulk Console window. See ["Processing bulk files" on page 280](#).

The user must log in to an Entrust desktop application for the change DN operation to complete.

You have now changed users' DN's in bulk by renaming the directory entry.

### To assign new DN's to users in bulk

The following procedure assumes that you have already changed the user's DN in the directory.

- 1 Open a new file in any text editor.

- 2 Identify the user to whom you want to assign a new DN by typing

```
user z "cn=Alice Gray,ou=Sales,dc=Company One,dc=com"
```

Substitute the DN in this statement with the old DN of the user to whom you want to assign a new DN. Remember to surround the user's DN with double quotation marks.

- 3 Assign the new DN to the user by typing

```
user_assigndn z "cn=Alice Jones,ou=Marketing,dc=Company \
One,dc=com"
```

Substitute `cn=Alice Jones, ou=Marketing, dc=Company One, dc=com` with the user's new DN. Remember to surround the user's DN with double quotation marks.

The backslash at the end of the first line keeps the information on the first and second lines together.

- 4 Repeat [Step 2](#) and [Step 3](#) for all the users to whom you want to assign a new DN.
- 5 Save the file with an `.entra` extension. The file can have any name (for example, `assigndn.entra`).
- 6 Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).

The user must log in to an Entrust desktop application for the assign DN operation to complete.

You have now assigned new DN's to users in bulk.



## Canceling the change DN operation in bulk

The change DN operation completes when users log in to an Entrust desktop application. If users have not yet logged in to an Entrust desktop application, you can cancel the change DN operation, if need be. However, before canceling the change DN operation, you must ensure a directory entry with the old DN exists. If the old entry does not exist, you must first rename the existing entry back to the old DN using the `user_renamedirectory` command.

If a directory entry with the old DN still exists, see [“To cancel the change DN operation in bulk” on page 313](#). If a directory entry with the old DN does not still exist, see [“To cancel the change DN operation in bulk when a directory entry with the old DN no longer exists” on page 314](#). If you are canceling a change DN operation and the old DN no longer exists in the directory (for example, you deleted it, or you used the `user_renamedirectory` command), then you must recreate the old DN entry in the directory before canceling the change DN operation.

---

**Note:** The following procedures shows sample bulk commands. Substitute the suggested variables and options according to your own requirements.

---

### To cancel the change DN operation in bulk

- 1 Open a new file in any text editor.
- 2 Identify the user for whom you want to cancel the change DN operation by typing

```
user z "cn=Alice Jones,ou=Product Management,dc=Company \
One,dc=com"
```

Substitute `cn=Alice Jones, ou=Product Management, dc=Company One, dc=com` with the DN of the user for whom you want to cancel the Change DN operation. In this example, the user is identified using a new DN. If you prefer, you can identify the user with the old DN, as in this example:

```
user z "cn=Alice Gray,ou=Engineering,dc=Company One,dc=com"
```

Remember to surround the user's DN with double quotation marks.

- 3 Cancel the DN change by typing  

```
user_cancelchangedn z
```
- 4 Repeat [Step 2](#) and [Step 3](#) for all the users for whom you want to cancel the Change DN operation.
- 5 Save the file with an `.entra` extension. The file can have any name (for example, `cancelchangedn.entra`).
- 6 Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).

You have now canceled the change DN operation in bulk.

### To cancel the change DN operation in bulk when a directory entry with the old DN no longer exists

- 1** Open a new file in any text editor.
  - 2** Identify the user for whom you want to cancel the change DN operation by typing  

```
user y "cn=Alice Jones,ou=Product Management,dc=Company \
One,dc=com"
```

Substitute "cn=Alice Jones, ou=Product Management, dc=Company One, dc=com" with the new DN of the user for whom you want to cancel the change DN operation. You must identify the user with the user's new DN. Remember to surround the user's DN with double quotation marks.
  - 3** Rename the user's directory entry with their old DN by typing  

```
user_renamedirentry y "cn=Alice Gray,ou=Engineering,dc=Company\
One,dc=com"
```

Substitute "cn=Alice Gray, ou=Engineering, dc=Company One, dc=com" with the user's old DN. Remember to surround the user's DN with double quotation marks.
  - 4** Cancel the DN change by typing  

```
user_cancelchangedn y
```
  - 5** Repeat [Step 2](#) to [Step 4](#) for all the users for whom you want to cancel the change DN operation.
  - 6** Save the file with an .entra extension. The file can have any name (for example, cancelchangedn.entra).
  - 7** Process this file in the Bulk Console window. See ["Processing bulk files" on page 280](#).
- You have now canceled the change DN operation in bulk.

## Changing user properties in bulk

Entrust PKI administrators can set or change a number of users' properties using the `user_setproperty` command. If you want to set the user's

- `subjectAltName`, see ["To set the user's subjectAltName component values in bulk" on page 315](#). The `subjectAltName` is the alternative identity. For more information on the `subjectAltName`, see ["Configuring subjectAltName values" on page 249](#).
- `key lifetimes`, see ["To set key lifetimes in bulk" on page 316](#). For more information on the `key lifetimes` property, see ["Configuring user key update options" on page 230](#).

- key expiry dates, see [“To set key expiry dates in bulk” on page 317](#). For more information on the key expiry dates property, see [“Configuring user key update options” on page 230](#).
- group membership, see [“To modify group membership in bulk” on page 318](#).
- role, see [“To modify the user’s role in bulk” on page 319](#).
- certificate category and type, see [“To set users’ certificate category and certificate type in bulk” on page 320](#).
- certificate extensions, see [“To set users’ certificate extensions in bulk” on page 321](#).

### To set the user’s subjectAltName component values in bulk

- 1 Open a new file in any text editor.
- 2 Identify the user whose subjectAltName component value you want to set by typing the following, for example:

```
user x "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute `cn=Bob Jones, dc=Company One, dc=com` with the DN of the user whose subjectAltName you want to set. Remember to surround the user’s DN with double quotation marks.

- 3 Set the subjectAltName component value by typing the following, for example:

```
user_setproperty x alternate_id +|- "<Component>=<Value>"
```

Where:

- `+|-` is used to add or remove component values.  
If you are removing component values and do not specify which component values you want deleted, everything is removed.
- `<Component>` is the subjectAltName component name.
- `<Value>` is the subjectAltName component value or values.  
For more information on determining the subjectAltName component names and values, see [Step 13 on page 291](#).

For example:

```
user_setproperty x alternate_id +  
"rfc822Name=bob.jones@company_one.com"
```

- 4 Apply the setting by typing  

```
user_applyproperties x
```
- 5 Update the user’s certificates.

If the user’s certificates are set for automatic key update, update the user’s key pairs using the `user_updatekeypairs` command. If the user’s certificates are set

for key expiry, then put the user into key recovery by using the `user_recover` command.

- 6** Repeat [Step 2](#) to [Step 5](#) for all the users whose `subjectAltName` component values you want to set.
- 7** Save the file with an `.entra` extension. The file can have any name (for example, `subaltname.entra`).
- 8** Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).

You have now set users' `subjectAltName` component values in bulk.

### To set key lifetimes in bulk

- 1** Open a new file in any text editor.
- 2** Identify the user whose key lifetimes you want to set by typing the following, for example:

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user whose key lifetimes you want to set. Remember to surround the user's DN with double quotation marks.

- 3** Set the user's key lifetimes by typing the following, for example:

```
user_setproperty z key_lifetimes 24 30 50
```

The values 24, 30, and 50 represent the encryption public key lifetime, the verification public key lifetime, and the signing private key lifetime respectively. Set these values according to your requirements. The encryption and verification public key lifetimes are set in terms of months, with a possible range of 2 to 60 months. The signing private key lifetime is set as a percentage of the verification public key lifetime, with a possible range of 1 to 100 percent.

Key lifetimes and key expiry dates are mutually exclusive. Setting key lifetimes overwrites any previously established key expiry dates. (The opposite is also true.) If you prefer to set key expiry dates, see [“To set key expiry dates in bulk” on page 317](#).

- 4** Apply the setting by typing
- 5** Update the user's certificates.

```
user_applyproperties z
```

If the user's certificates are set for automatic key update, update the user's key pairs using the `user_updatekeypairs` command. If the user's certificates are set for key expiry, then put the user into key recovery by using the `user_recover` command.

- 6** Repeat [Step 2](#) to [Step 5](#) for all the users whose key lifetimes you want to set.

- 7 Save the file with an `.entra` extension. The file can have any name (for example, `keylifetimes.entra`).
- 8 Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).  
You have now set users' key lifetimes in bulk.

### To set key expiry dates in bulk

- 1 Open a new file in any text editor.
- 2 If you have not already done so, set the user up for key recovery. See [“Setting up users for key recovery in bulk” on page 298](#).
- 3 Identify the user whose key expiry dates you want to set by typing the following, for example:

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user whose key expiry dates you want to set. Remember to surround the user's DN with double quotation marks.

- 4 Set the user's key expiry dates by typing the following, for example:

```
user_setproperty z key_expiry "12/01/2010 7:00:00 AM" "12/31/2010 7:00:00 AM"
```

The value `"12/01/2010 7:00:00 AM"` represents the expiry date of both the user's encryption public key and the user's signing private key. The value `"12/31/2010 7:00:00 AM"` represents the expiry date of the user's verification public key. Set these values according to your requirements.

The format for the date is dependent on regional settings. For example, the default for English (US) is `MM/DD/YYYY`. Enter the time as `hh:mm:ss` followed by `AM` or `PM`.

The date and time must occur at least 12 hours into the future. The date and time cannot exceed the expiry date of the CA's signing certificate. The date and time contains spaces so you must enclose the date and time in quotation marks.

Key expiry dates and key lifetimes are mutually exclusive. Setting key expiry dates overwrites any previously established key lifetimes. (The opposite is also true.) If you prefer to set key lifetimes, see [“To set key lifetimes in bulk” on page 316](#).

- 5 Apply the setting by typing  

```
user_applyproperties z
```
- 6 Update the user's certificates.

If the user's certificates are set for automatic key update, update the user's key pairs using the `user_updatekeypairs` command. If the user's certificates are set for key expiry, then put the user into key recovery by using the `user_recover` command.

- 7 Repeat [Step 3](#) to [Step 6](#) for all the users whose key expiry dates you want to set.
- 8 Save the file with an `.entra` extension. The file can have any name (for example, `keyexpirydates.entra`).
- 9 Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).

You have now set users' key expiry dates in bulk.

### To modify group membership in bulk

- 1 Open a new file in any text editor.
- 2 Identify the user whose group membership you want to modify by typing the following, for example:

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user whose group membership you want to modify. Remember to surround the user's DN with double quotation marks.

- 3 If you want to add the user to a group, go to [Step 4](#). If you want to remove the user from a group, go to [Step 5](#).
- 4 Add the user to a group by typing the following, for example:

```
user_setproperty z group + "Group A"
```

Substitute `Group A` with the name of the group you want to add the user to. The group must already exist before you can add the user to it.

---

**Note:** Double quotation marks must surround any group names of more than one word.

---

To add the user to numerous groups, include a `user_setproperty` command for each group you are adding the user to. For example:

```
user_setproperty z group + "Group B"
```

```
user_setproperty z group + "Group C"
```

To add the user to all groups that currently exist as well as to any future groups, type

```
user_setproperty z group + *
```

To remove the user from a group or groups, go to [Step 5](#). Otherwise, proceed to [Step 6](#).

- 5 Remove the user from a group by typing the following, for example:

```
user_setproperty z group - "Group K"
```

Substitute `Group K` with the name of the group you want to remove the user from. The user must always remain a member of at least one group.

To remove the user from numerous groups, include a `user_setproperty` command for each group you are removing the user from. For example:

```
user_setproperty z group - "Group B"
user_setproperty z group - "Group C"
```

To remove the user from all groups, type the following:

```
user_setproperty z group - *
```

A user must belong to at least one group. To add the user to a group or groups, go to [Step 4](#).

**6** Apply the setting by typing

```
user_applyproperties z
```

**7** Update the user's certificates.

If the user's certificates are set for automatic key update, update the user's key pairs using the `user_updatekeypairs` command. If the user's certificates are set for key expiry, then put the user into key recovery by using the `user_recover` command.

**8** Repeat [Step 2](#) to [Step 7](#) for all the users whose group membership you want to modify.

**9** Save the file with an `.entra` extension. The file can have any name (for example, `addgroups.entra`).

**10** Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).

You have now modified users' group membership in bulk.

### To modify the user's role in bulk

**1** Open a new file in any text editor.

**2** Identify the user whose role you want to modify by typing the following, for example:

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user whose role you want to modify. Remember to surround the user's DN with double quotation marks.

**3** Modify the user's role by typing

```
user_setproperty z role "End User"
```

You can also create an Entrust PKI administrator by substituting `"End User"` with `"Security Officer"`, `"Administrator"`, `"Directory Administrator"`, `"Auditor"`, or any other role, as required. You must surround role names made up of two or more words (End User, Security Officer, Directory Administrator) with double quotation marks.

**4** Apply the modification by typing

```
user_applyproperties z
```

**5** Update the user's certificates.

If the user's certificates are set for automatic key update, update the user's key pairs using the `user_updatekeypairs` command. If the user's certificates are set for key expiry, then put the user into key recovery by using the `user_recover` command.

**6** Repeat [Step 2](#) to [Step 5](#) for all the users whose roles you want to modify.

**7** Save the file with an `.entra` extension. The file can have any name (for example, `changerole.entra`).

**8** Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).

You have now modified users' roles in bulk.

### To set users' certificate category and certificate type in bulk

**1** Open a new file in any text editor.

**2** Identify the user whose certificate category you want to set by typing the following, for example:

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user whose certificate category and type you want to set. Remember to surround the user's DN with double quotation marks.

**3** Set the user's certificate category and type by typing the following, for example:

```
user_setproperty z certificate_type Web web_default
```

Substitute `Web` in this statement with the certificate category you want to assign to the user. Your certificate category options are `Enterprise` or `Web`. Note that you can only assign a Web certificate to a user if you have purchased licenses for Web certificates.

Substitute `web_default` in the above statement with the appropriate certificate type. To see what certificate categories and types are available to you, log in to Security Manager Administration as an Entrust PKI administrator and click **Users > New User**. Then follow the steps in [“Configuring user certificate types” on page 221](#). When you are done viewing the certificate categories and types, click **Cancel** and continue with the rest of this procedure.

**4** Apply the setting by typing

```
user_applyproperties z
```

**5** Update the user's certificates.

If the user's certificates are set for automatic key update, update the user's key pairs using the `user_updatekeypairs` command. If the user's certificates are set



for key expiry, then put the user into key recovery by using the `user_recover` command.

- 6** Repeat [Step 2](#) to [Step 5](#) for all the users whose certificate category and type you want to set.
- 7** Save the file with an `.entra` extension. The file can have any name (for example, `certcategories.entra`).
- 8** Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).  
You have now set users' certificate category and type in bulk.

### To set users' certificate extensions in bulk

- 1** Open a new file in any text editor.
- 2** Identify the user whose certificate extensions you want to set by typing the following, for example:

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user whose certificate extensions you want to set. Remember to surround the user's DN with double quotation marks.

- 3** Set the user's certificate extensions by typing the following, for example:

```
user_setproperty z certificate_extension userStreet "100 East\  
27th Street"
```

In the above example, the certificate extension is `userStreet`, and the value is `"100 East 27th Street"`.

The backslash at the end of the first line keeps the information on the two lines together. Without the backslash, the two lines are interpreted as separate statements. The double quotation marks indicate that all the words inside the double quotation marks are a single argument.

To specify a certificate extension, you must use the extension's Var ID, as defined in the `master.certspec` file. For more information on certificate extensions and the `master.certspec` file, see [“Customizing certificates” on page 525](#).

Here are some other examples of setting certificate extensions:

```
user_setproperty z certificate_extension userCity "New York"  
user_setproperty z certificate_extension userEmail \  
bob@company_one.com  
user_setproperty z certificate_extension policyOIDS \  
"2.6.7.1.47.98.2 2.6.7.1.47.98.4"
```

In the first of these command statements, `"New York"` is the value assigned to the certificate extension `userCity`. The double quotation marks around New York indicate that the words are treated as a single argument.

In the second command statement, "bob@company\_one.com" is the extension assigned to the certificate extension `userEmail`. No double quotation marks are required because `bob@company_one.com` appears as a single unbroken word.

In the third command statement, the certificate extension policyOIDs contains two values: `2.6.7.1.47.98.2` and `2.6.7.1.47.98.4`. When a certificate extension requires multiple values, always specify the values as a space-separated list enclosed in double quotation marks.

**4** Apply the setting by typing

```
user_applyproperties z
```

**5** Update the user's certificates.

If the user's certificates are set for automatic key update, update the user's key pairs using the `user_updatekeypairs` command. If the user's certificates are set for key expiry, then put the user into key recovery by using the `user_recover` command.

**6** Repeat [Step 2](#) to [Step 5](#) for all the users whose certificate extensions you want to specify.

**7** Save the file with an `.entra` extension. The file can have any name (for example, `certextensions.entra`).

**8** Process this file in the Bulk Console window. See ["Processing bulk files" on page 280](#).

You have now specified users' certificate extensions in bulk.

## Adding and deleting directory attributes in bulk

Use the `user_setattribute` command to add directory attributes when you create a new user in the directory. This is the same command used to add and delete directory attributes for existing users.

The `user_setattribute` command creates new attributes and adds values to existing attributes. If an attribute is already assigned a value, new values added to the attribute are added to the existing list of values.

The `user_setattribute` command also deletes attribute values and attributes. If an attribute has only one value, then deleting that value also deletes the attribute.

You can also use the `user_setattribute` command to add and delete directory attributes for entries in your directory that are not related to Security Manager.

For more information, see [“Using the Directory Browser” on page 69](#).

### To add and delete directory attributes in bulk

- 1 Open a new file in any text editor.
- 2 Identify the user whose directory attribute you want to add, delete, or modify by typing the following, for example:

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user you want to modify. Remember to surround the user's DN with double quotation marks.

- 3 If you want to add a directory attribute, or set an attribute value, go to [Step 4](#). If you want to delete a directory attribute or remove an attribute value, go to [Step 5](#).

- 4 Add a directory attribute by typing the following, for example:

```
user_setattribute z FavoriteFoods + Potatoes
```

Substitute `FavoriteFoods` with the name of the directory attribute you want to create. Substitute `Potatoes` with the value you want to assign to the new directory attribute.

You can also add values to a directory attribute that already exists by typing the following, for example:

```
user_setattribute z FavoriteFoods + Pasta
```

```
user_setattribute z FavoriteFoods + Porridge
```

Substitute `FavoriteFoods` with the name of the directory attribute you are adding values to. Substitute `Pasta` and `Porridge` with the values you want to add to the given directory attribute.

- 5 Delete a directory attribute by typing the following, for example:

```
user_setattribute z FavoriteFoods - *
```

Substitute `FavoriteFoods` with the name of the directory attribute you want to delete. This bulk command statement removes the given directory attribute and all values assigned to this attribute.

If you want to remove a specific value from the directory attribute, type the following, for example:

```
user_setattribute z FavoriteFoods - Porridge
```

Substitute `FavoriteFoods` with the name of the directory attribute you want to remove a value from. Substitute `Porridge` with the value you want to remove.

---

**Note:** Removing all values from the attribute deletes the directory attribute itself.

---

- 6** Repeat [Step 2](#) to [Step 5](#) for all the users whose directory attributes you want to modify.
- 7** Save the file with an `.entra` extension. The file can have any name (for example, `addattribute.entra`).
- 8** Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).

You have now modified users' directory attributes in bulk.

# Restoring information to the directory in bulk

You can restore a user's certificates to the directory if the certificates are accidentally deleted or corrupted.

You should back up your directory on a regular basis using your directory backup tools. The procedure described in this section only restores certificate information. If any other type of directory information becomes corrupt or is lost, such as object classes, directory entries, or directory attributes, you can only retrieve this information from a backup generated by your directory backup tools. For more information about backing up your directory, see the *Security Manager Operations Guide*.

## To restore users' certificates to the directory in bulk

- 1 Open a new file in any text editor.
- 2 Identify the user whose certificates you want to restore by typing the following, for example:

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user whose certificates you want to restore. Remember to surround the user's DN with double quotation marks.

- 3 Restore the user's certificates by typing  

```
user_restoretodir z
```
- 4 Repeat [Step 2](#) and [Step 3](#) for all the users whose certificates you want to restore.
- 5 Save the file with an `.entra` extension. The file can have any name (for example, `restorecerts.entra`).
- 6 Process this file in the Bulk Console window. See ["Processing bulk files" on page 280](#).

You have now restored users' certificates in bulk.

# Updating users' key pairs in bulk

For general information about updating key pairs, see ["Updating key pairs" on page 240](#).

---

**Note:** If you have set the user encryption key to RSA-4096 or RSA-6144, the key update operation is considerably slower than for RSA-1024 or RSA-2048. See the *Security Manager Operations Guide* for more information.

---

## To update users' key pairs in bulk

- 1** Open a new file in any text editor.
- 2** Identify the user whose key pairs you want to update by typing the following, for example:  

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user whose key pairs you want to update. Remember to surround the user's DN with double quotation marks.
- 3** Update the user's key pairs by typing  

```
user_updatekeypairs z
```
- 4** Repeat [Step 2](#) and [Step 3](#) for all the users whose key pairs you want to update.
- 5** Save the file with an `.entra` extension. The file can have any name (for example, `updatekeypairs.entra`).
- 6** Process this file in the Bulk Console window. See ["Processing bulk files" on page 280](#).  
You have now updated users' key pairs in bulk.

# Notifying client applications in bulk

For general information about notifying users' clients, see [“Notifying client applications in bulk” on page 327](#).

## To notify client applications in bulk

- 1 Open a new file in any text editor.
- 2 Identify the user whose clients you want to notify by typing the following, for example:

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user whose client you want to notify. Remember to surround the user's DN with double quotation marks.

- 3 Notify the user's client by typing  

```
user_notifyclient z
```
- 4 Repeat [Step 2](#) and [Step 3](#) for all the users whose clients you want to notify.
- 5 Save the file with an `.entra` extension. The file can have any name (for example, `notifyclients.entra`).
- 6 Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).

You have now notified users' clients in bulk.

# Reissuing activation codes in bulk

Reissuing activation codes generates new codes with new creation and expiry dates. The validity of the previous activation codes does not affect this procedure. You can reissue new activation codes before the old activation codes expire (for example, when a user's activation codes are lost or stolen). For more information about activation code lifetimes, see [“Configuring the lifetime of activation codes” on page 154](#).

If the Entrust PKI administrator who processes the bulk command file has permission to view activation codes, the activation codes appear in the bulk log output file.

## To reissue activation codes in bulk

- 1** Open a new file in any text editor.
- 2** Identify the user whose activation codes you want to reissue by typing the following, for example:  

```
user z "cn=Bob Jones,dc=Company One,dc=com"
```

Substitute the DN in this statement with the DN of the user whose activation codes you want to reissue. Remember to surround the user's DN with double quotation marks.
- 3** Reissue the user's activation codes by typing  

```
user_reissueactivationcodes z
```
- 4** Repeat [Step 2](#) and [Step 3](#) for all the users whose activation codes you want to reissue.
- 5** Save the file with an `.entra` extension. The file can have any name (for example, `actcodes.entra`).
- 6** Process this file in the Bulk Console window. See [“Processing bulk files” on page 280](#).  
You have now reissued users' activation codes in bulk.



## Administering groups

Groups help you organize and locate users. While searchbases fulfill a similar function (see [“Administering searchbases” on page 341](#)), searchbases follow a strict directory structure. Groups are completely flexible and are not dependent on the directory structure. You can create a group and add a user to the group regardless of where the user is listed in the directory. Organizing users into groups can improve the security and efficiency of your Entrust PKI system.

Assigning users to different groups allows you to impose administrative restrictions on different groups, improving security. For example, you can divide your organization into different groups such as Sales and Marketing, and then configure different Entrust PKI administrators so they can manage only the Sales or Marketing groups. Highly-privileged administrators such as Security Officers can administer all groups.

Groups also increase the efficiency of searches and reports. Rather than searching through the entire database to find one user, you can limit your searches to a particular group.

Entrust PKI administrators with sufficient permissions can administer groups. By default, only Security Officers can administer groups. For more information about roles, see [“Administering roles” on page 353](#).

Topics in this section:

- [“Viewing groups” on page 330](#)
- [“Creating groups” on page 332](#)
- [“Adding members to groups” on page 334](#)
- [“Removing members from groups” on page 336](#)
- [“Renaming groups” on page 338](#)
- [“Deleting groups” on page 339](#)

# Viewing groups

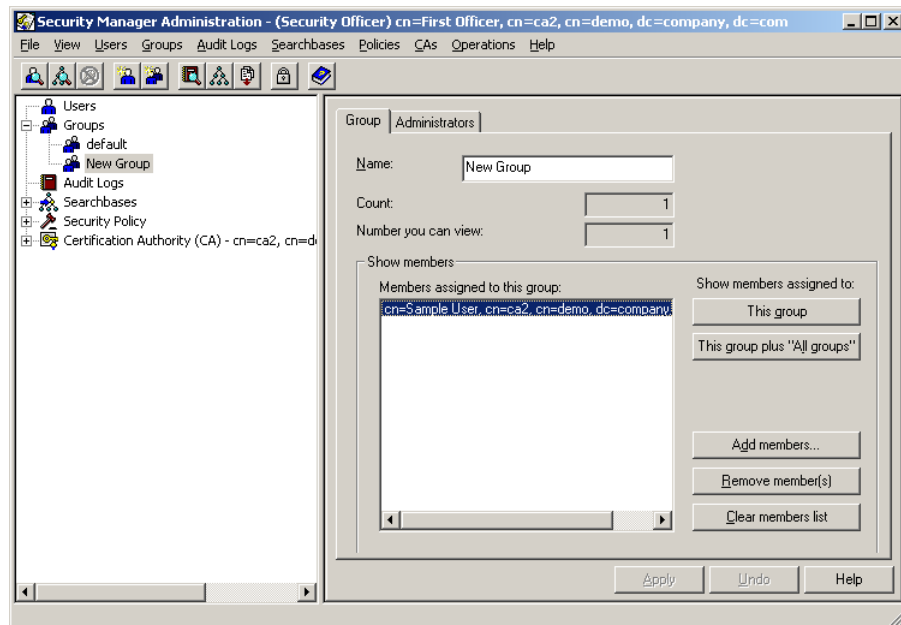
Entrust PKI administrators with sufficient permissions can view groups in Security Manager Administration. Entrust PKI administrators can also view a list of administrators that can administer a selected group. Entrust PKI administrators can only view the groups that their own role allows them to view. See [“Permissions reference” on page 371](#) for a list of all administrative permissions.

This section contains the following procedures:

- [“To view a group” on page 330](#)
- [“To view a list of administrators that can administer a group” on page 331](#)

## To view a group

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).



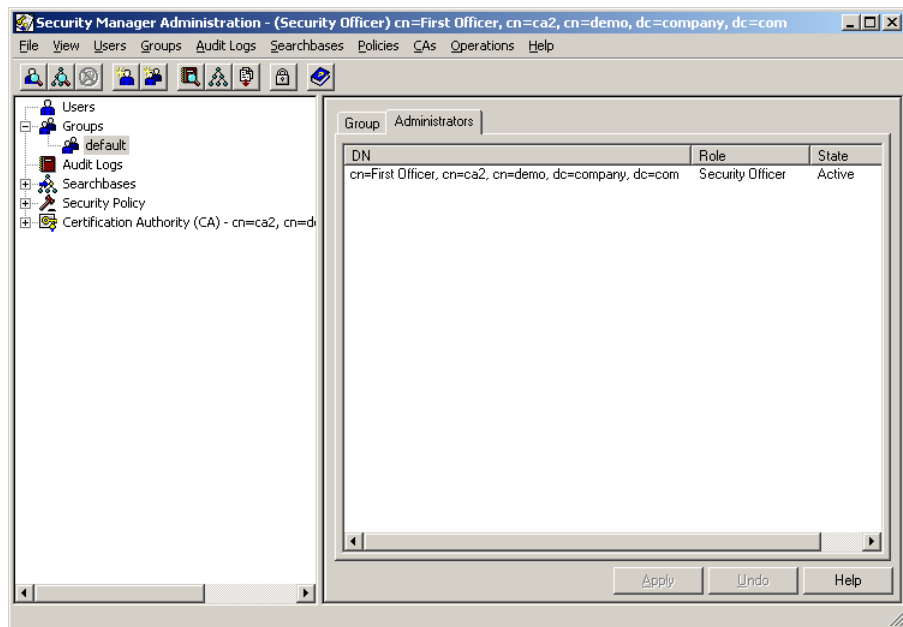
- 2 In the tree view, expand **Groups**.  
A list of groups appears. The groups that appear depends on the list of groups that your own role allows you to access.
- 3 Select the group that you want to view.
- 4 Click the **Group** tab.
- 5 To view a list of users assigned to the group, click **This group**.  
A list of group members appear in the **Members assigned to this group** box.

- 6 To view users assigned to the group as well as users assigned to all groups, click **This group plus "All groups"**.

A list of group members and users assigned to all groups appear in the **Members assigned to this group** box. Users with an asterisk (\*) beside their name are users assigned to all groups.

### To view a list of administrators that can administer a group

- 1 Log in to Security Manager Administration. See ["Logging in to Security Manager Administration" on page 46](#).



- 2 In the tree view, expand **Groups**.  
A list of groups appears. The groups that appear depends on the list of groups that your own role allows you to access.
- 3 Select the group you want to view.
- 4 Click the **Administrators** tab to view the list of all Entrust PKI administrators who can administer the selected group.

# Creating groups

By default, each user must belong to at least one group. Security Manager includes a default group. If you created no other groups, all new users are added to the default group. Entrust PKI administrators with sufficient permissions can create new groups.

## To create a group

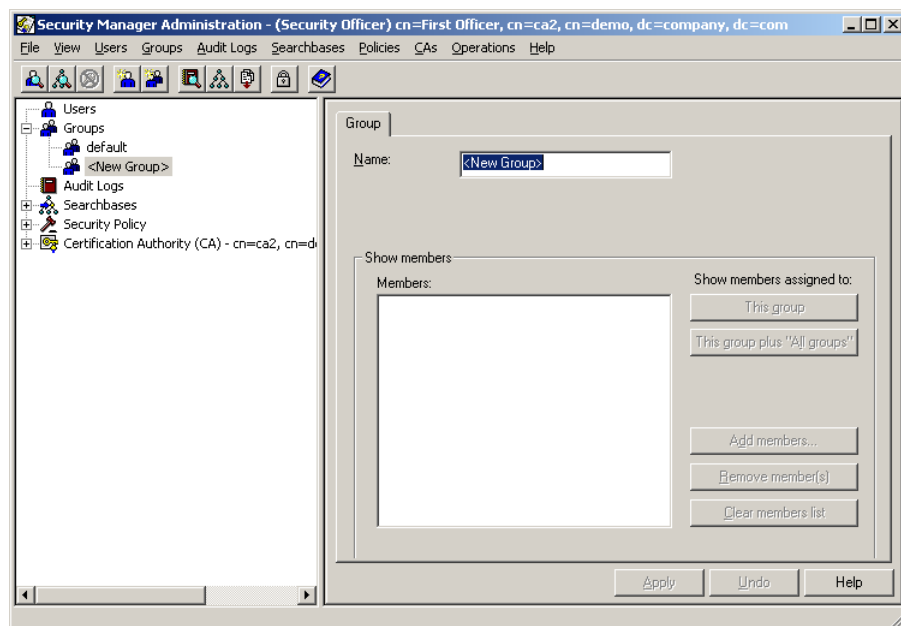
- 1 Log in to Security Manager Administration. See [“Logging in to Security Manager Administration” on page 46.](#)
- 2 Select **Groups > New Group**.

---

**Note:** If you start to create a new group and then change your mind, you can back out of the operation by selecting **Groups > Refresh List**.

---

A **<New Group>** entry appears below the **Groups** icon, and the **Group** property page appears in the right pane of Security Manager Administration.



- 3 In the **Name** field, enter a name for the new group.  
You must enter a unique name for the group. If you enter an existing group name, Security Manager Administration prompts you to enter a different name.
- 4 Click **Apply**.
- 5 If prompted, authorize the operation. See [“Authorizing sensitive operations” on page 52.](#)

If the operation was successful, a success message appears.

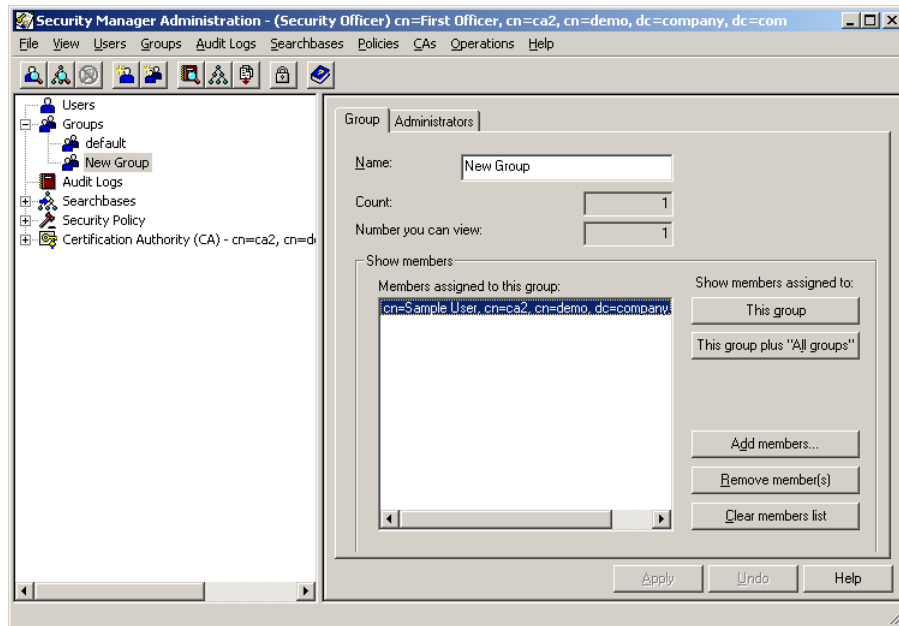
# Adding members to groups

Users must belong to at least one group. Entrust PKI administrators with sufficient permissions can add members to groups.

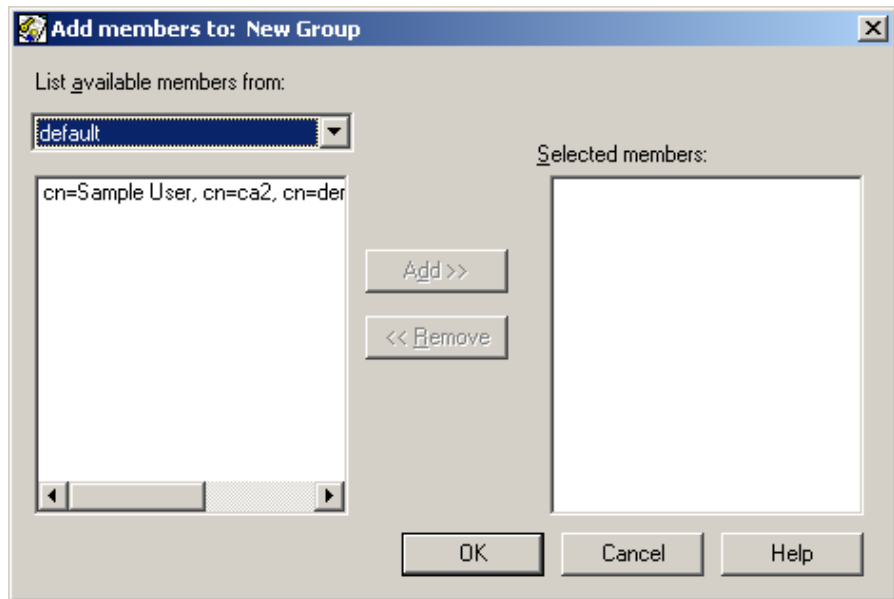
This section describes how to add members to a group. Alternatively, you can modify a user to change the user's group membership. See [“Configuring user properties” on page 219](#) for details.

## To add members to a group

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).



- 2 In the tree view, expand **Groups**.  
A list of groups appears. The groups that appear depends on the list of groups that your own role allows you to access.
- 3 Select the group that you want to add users to.
- 4 Click the **Group** tab.
- 5 Click **Add members**.  
The **Add members to** dialog box appears.



- 6 To add a member:
  - a In the **List available members from** drop-down list, select a group that contains the member that you want to add.

At least three groups must exist for the drop-down list to appear. If only two groups exist, you can only add members from the other group.

The members of the selected group appear in the **List available members from** box.
  - b Select a member from the list and then click **Add**.

The member appears in the **Selected members** list.
  - c Repeat [Step 6](#) as necessary
  - d and click **OK**.
- 7 Click **Apply**.
- 8 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

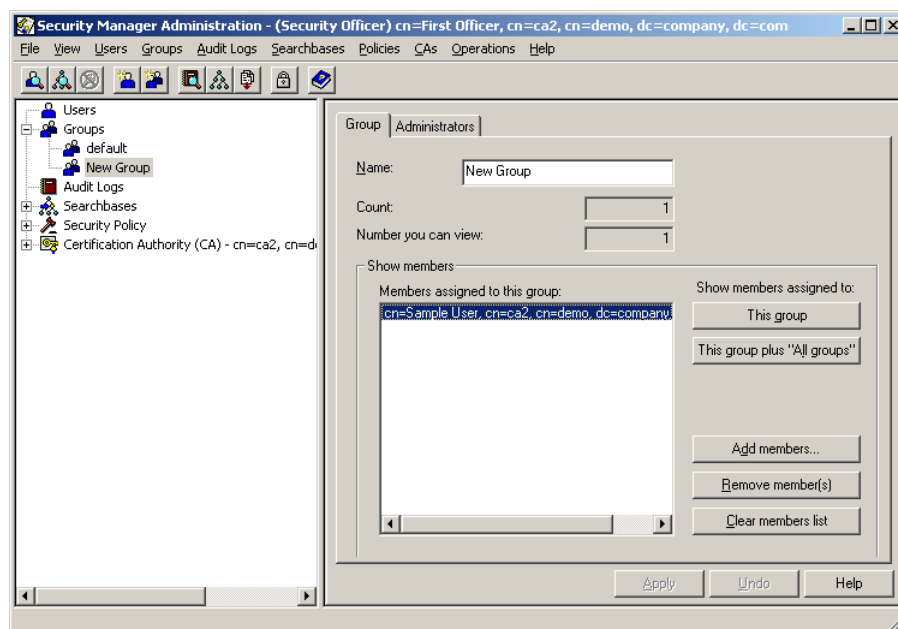
# Removing members from groups

You can remove a member from a group as long as the member belongs to another group. Entrust PKI administrators with sufficient permissions can add members to groups.

This section describes how to remove members from a group. Alternatively, you can modify a user to change the user's group membership. See ["Configuring user properties" on page 219](#) for details.

## To remove members from a group

- 1 Log in to Security Manager Administration (see ["Logging in to Security Manager Administration" on page 46](#)).



- 2 In the tree view, expand **Groups**.  
A list of groups appears. The groups that appear depends on the list of groups that your own role allows you to access.
- 3 Select the group that contains the members you want to remove.
- 4 Click the **Group** tab.
- 5 Click **This Group** to display all members belonging to this group.
- 6 To remove members:
  - To remove an individual member, select the member from the **Members assigned to this group** list and then click **Remove member**.



The member is removed from the **Members assigned to this group** list.

- To remove all members from the group, click Clear members list.

All members are removed from the **Members assigned to this group** list.

**7** Click **Apply**.

**8** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

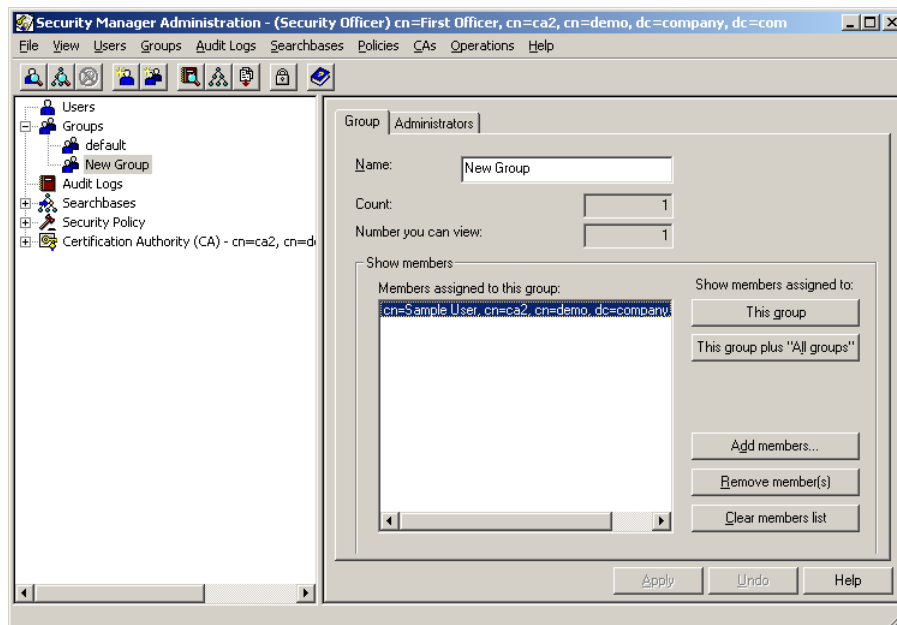
If the operation was successful, a success message appears.

# Renaming groups

Entrust PKI administrators with sufficient permissions can rename groups. When you rename a group, the name change occurs globally. If you try to rename a group that you cannot rename (for example, you do not have sufficient permissions to rename the group), a message appears explaining why you cannot rename the group.

## To rename a group

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration”](#) on page 46).



- 2 In the tree view, expand **Groups**.  
A list of groups appears. The groups that appear depends on the list of groups that your own role allows you to access.
- 3 Select the group that you want to rename.
- 4 Click the **Group** tab.
- 5 In the **Name** field, enter a new name for the group.
- 6 Click **Apply**.
- 7 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations”](#) on page 52.  
If the operation was successful, a success message appears.

# Deleting groups

Entrust PKI administrators with sufficient permissions can delete groups. Delete a group if your organization no longer requires the group, or if you created a group by mistake.

User must belong to at least one group. You can delete groups that currently contain members. If you attempt to delete a group that contains members and at least one of those members belongs only to that group, an error occurs.

Before deleting groups, ensure that each user in the group belongs to at least one other group. It is recommended that you remove all users from the group before deleting the group (see [“Removing members from groups” on page 336](#)).

You can delete the default group if you have at least one other group and all Entrust users belong to the other group.

## To delete a group

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** In the tree view, expand **Groups**.  
A list of groups appears. The groups that appear depends on the list of groups that your own role allows you to access.
- 3** Select the group you want to delete.
- 4** Select **Groups > Select Group > Delete**.  
A confirmation dialog box appears requesting that you confirm the action.
- 5** Click **OK** to continue.
- 6** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
If the operation was successful, a success message appears.



## Administering searchbases

Searchbases help you organize and locate users. While groups fulfill a similar function (see [“Administering groups” on page 329](#)), searchbases follow a strict directory structure. Searchbases generally follow logical administrative lines in an organization, typically by organizational unit (see [“Adding searchbases” on page 344](#)).

Searchbases improve the efficiency of finding users in a large Certification Authority (CA) domain or in cross-certified domains. A CA domain is a collection of users who all use Security Manager under the same software license and who are certified by the same CA. If you do not add searchbases, all users you add are located under the CA domain. The default CA domain is called CA Domain Searchbase.

When two CAs are cross-certified, each CA must create a searchbase for the other domain so that users can exchange secured files across the domains. A searchbase is the only means available to find recipients in a cross-certified domain. To search for recipients in a cross-certified domain, Entrust desktop application users must select the searchbase representing that domain.

---

**Note:** Creating a directory structure and operating a directory is usually a task assigned to an Entrust PKI administrator with sufficient permissions (see [“Administering roles” on page 353](#)). If you have any questions about creating directory entries, contact this individual in your organization.

---

This chapter contains the following sections:

- [“Viewing searchbases” on page 342](#)
- [“Adding searchbases” on page 344](#)
- [“Modifying searchbases” on page 349](#)
- [“Deleting searchbases” on page 351](#)

# Viewing searchbases

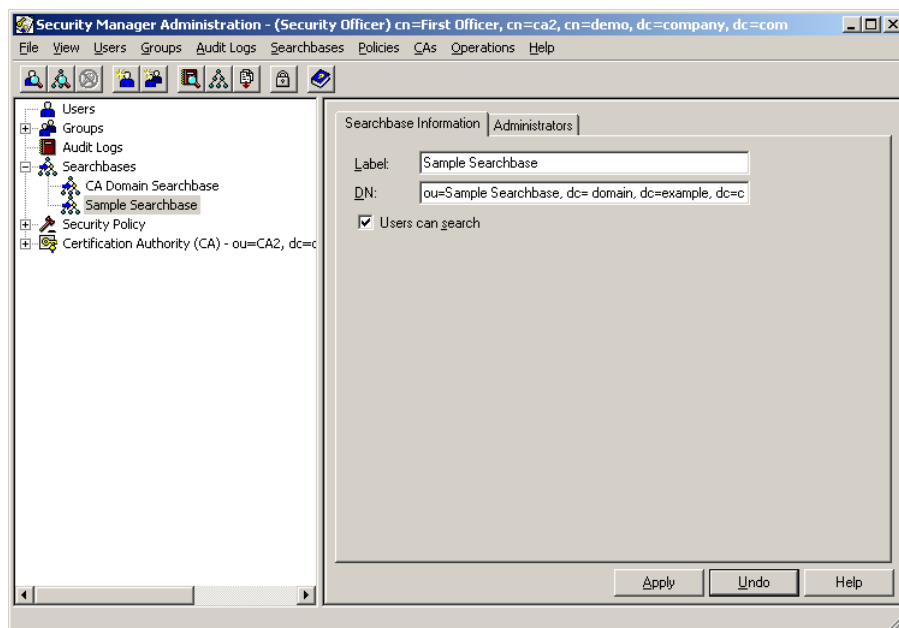
Entrust PKI administrators with sufficient permissions can view searchbases in Security Manager Administration. Entrust PKI administrators can also view a list of administrators that can administer a selected searchbase. Entrust PKI administrators can only view the searchbases that their own role allows them to view. See [“Permissions reference” on page 371](#) for a list of all administrative permissions.

This section contains the following procedures:

- [“To view a searchbase and its properties” on page 342](#)
- [“To view a list of administrators that can administer a searchbase” on page 343](#)

## To view a searchbase and its properties

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).

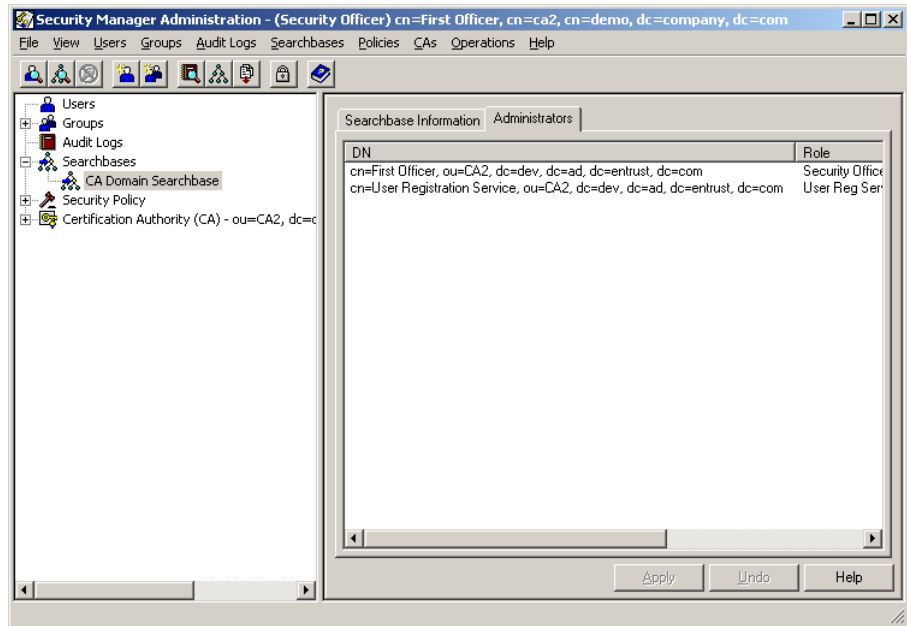


- 2 In the tree view, expand **Searchbases**.  
A list of searchbases appears. The searchbases that appear depends on the list of searchbases that your own role allows you to access.
- 3 Select the searchbase that you want to view.

The **Searchbase Information** property page appears in the right pane of Security Manager Administration. The property page displays the label and DN of the searchbase.

## To view a list of administrators that can administer a searchbase

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration”](#) on page 46).

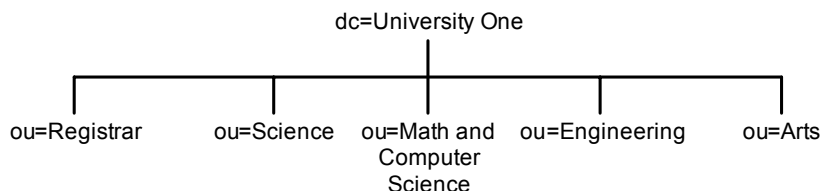


- 2 In the tree view, expand **Searchbases**.  
A list of searchbases appears. The searchbases that appear depends on the list of searchbases that your own role allows you to access.
- 3 Select the searchbase that you want to view.
- 4 Click the **Administrators** tab to view the list of all Entrust PKI administrators who can administer the selected role.

# Adding searchbases

Searchbases help you organize and locate users. While groups fulfill a similar function (see [“Administering groups” on page 329](#)), searchbases follow a strict directory structure. Searchbases generally follow logical administrative lines in an organization, typically by organizational unit (see Figure 6).

**Figure 6:** An example of a Directory Information Tree



Searchbases improve the efficiency of finding users in a large Certification Authority (CA) domain or in cross-certified domains. A CA domain is a collection of users who all use Security Manager under the same software license and who are certified by the same CA. If you do not add searchbases, all users you add are located under the CA domain. The default CA domain is called CA Domain Searchbase.

---

**Note:** Creating a directory structure and operating a directory is usually a task assigned to an Entrust PKI administrator with sufficient permissions (see [“Administering roles” on page 353](#)). If you have any questions about creating directory entries, contact this individual in your organization.

---

When two CAs are cross-certified, each CA must create a searchbase for the other domain so that users can exchange secured files across the domains. A searchbase is the only means available to find recipients in a cross-certified domain. To search for recipients in a cross-certified domain, Entrust desktop application users must select the searchbase representing that domain.

For example, if your CA domain, `dc=Company One,dc=com`, is cross-certified with the CA domain `dc=Company Two,dc=com`, you must add the searchbase `dc=Company Two,dc=com` so users in your domain can access the encryption certificates of users in the other domain. Likewise, an administrator in the Company Two domain must create a searchbase for `dc=Company One,dc=com` (your CA domain) so that users in the Company Two domain can access recipients in your domain.

Adding searchbases is a two-step process:

- 1 Add a searchbase to the directory (see [“Adding searchbases to the directory” on page 345](#)).



Before you do this, however, make sure that the entry on which you want to base the searchbase already exists in the directory. (Check with a Directory Administrator, or someone else who is familiar with the directory structure of your organization.) The following procedures model searchbases after organizational units.

- 2 Add the searchbase to Security Manager (see [“Adding searchbases to Security Manager” on page 347](#)).

Only Entrust PKI administrators with sufficient permissions can add searchbases (see [“Administering roles” on page 353](#)).

This section contains the following topics:

- [“Adding searchbases to the directory” on page 345](#)
- [“Adding searchbases to Security Manager” on page 347](#)

## Adding searchbases to the directory

Before you can add a searchbase to Security Manager, you must add the searchbase to the directory.

By default, you can create searchbases using only organizational units (ou=). You can create searchbases using other directory components such as locality (l=) or country (c=), but you must structure your directory accordingly. Check with a Directory Administrator or someone else who is familiar with the directory structure of your organization before creating searchbases in your directory.

If you use Active Directory, you cannot add a searchbase using the Directory Browser, you must use your Active Directory tools. If you use another supported directory, you do not have to use the Directory Browser, you can use your directory tools if required.

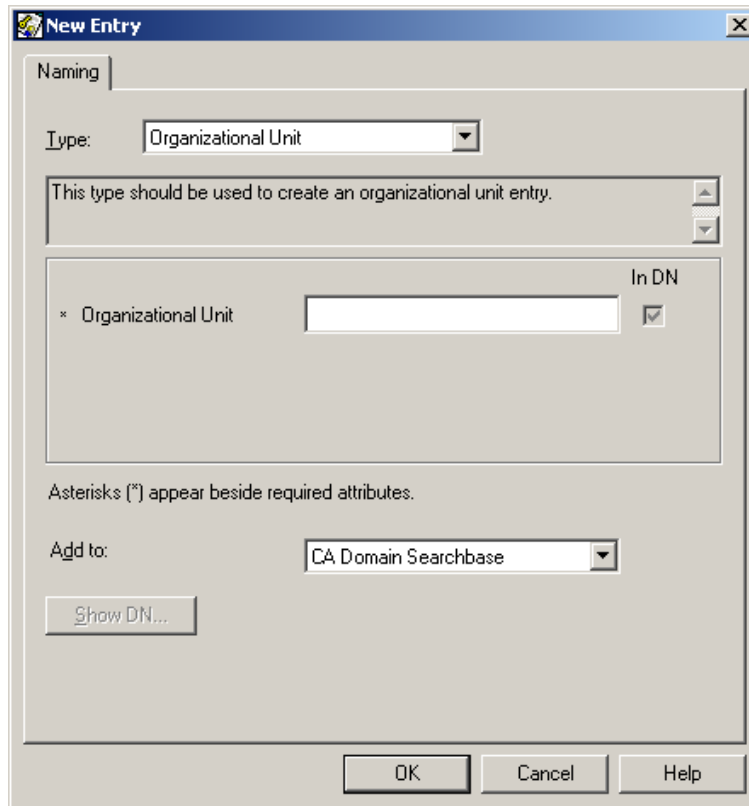
---

**Note:** Creating a directory structure and operating a directory is usually a task assigned to an Entrust PKI administrator with sufficient permissions (see [“Administering roles” on page 353](#)). If you have any questions about creating directory entries, contact this individual in your organization.

---

### To add an organizational unit to the directory

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Select **Operations > Directory Browser**.  
The Directory Browser appears.
- 3 Select **Entries > New Entry**.  
The **New Entry** dialog box appears.



- 4** In the **Type** drop-down list, select **Organizational Unit**.
- 5** In the **Organizational Unit** field, enter the name of the organizational unit.
- 6** In the **Add to** drop-down list, select the existing searchbase where you want to add the new searchbase.  
 The existing searchbase under which you locate the organizational unit depends on the Directory Information Tree (DIT) structure of your organization. Do not assume that the searchbase must reside below the **CA Domain Search Base**. Contact a Directory Administrator about where to position the organizational unit if you have any doubts about the DIT structure.
- 7** Click **Show DN** to preview the distinguished name of the searchbase. If required, change the name of the organizational unit or the existing searchbase.
- 8** Click **OK**.
- 9** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears. You can now create add the searchbase to Security Manager (see [“Adding searchbases to Security Manager” on page 347](#)).

## Adding searchbases to Security Manager

After adding a searchbase to the directory (see [“Adding searchbases to the directory” on page 345](#)), you can add the searchbase to Security Manager. After adding a searchbase to Security Manager, Entrust PKI administrators can add users to the searchbase.

Only Entrust PKI administrators with sufficient permissions can add searchbases to Security Manager.

When you add a searchbase to Security Manager, you include a label and a distinguished name (DN) for the searchbase. This information goes into the Security Manager database. The DN then maps to a particular entry in the Directory Information Tree (DIT). If there is no corresponding directory entry, it cannot establish the searchbase.

### To add a searchbase

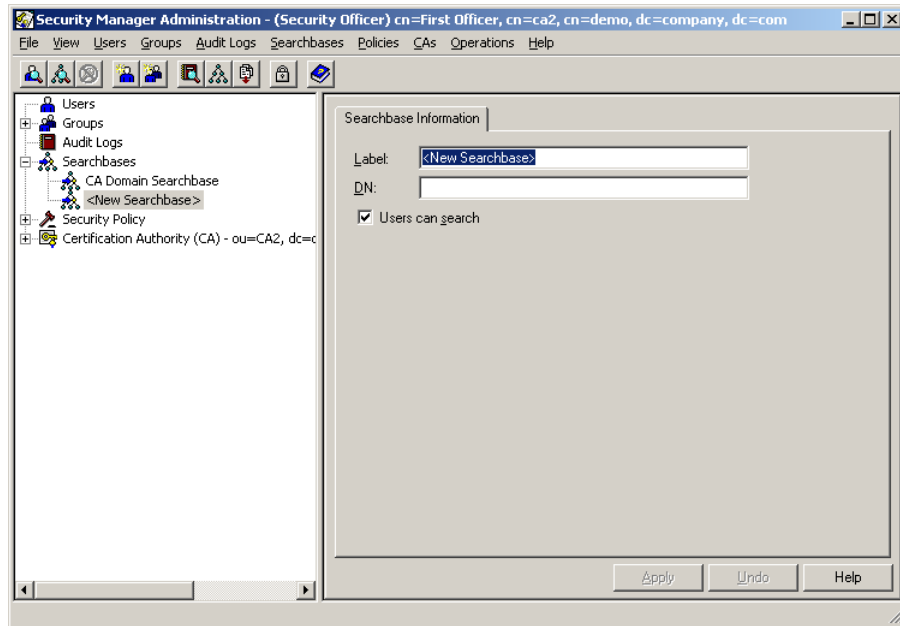
- 1** Before you can add a searchbase, you must add a searchbase to the directory. See [“Adding searchbases to the directory” on page 345](#) for details.
- 2** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 3** Select **Searchbases > New Searchbase**.

A searchbase with the name **<New Searchbase>** appears in the tree view. The **Searchbase Information** appears in the right pane.

---

**Note:** If you start to create a new searchbase and then change your mind, you can back out of the operation by selecting **Searchbases > Refresh**.

---



- 4 In the **Label** field, enter a unique name for the searchbase.  
The name you enter is the one users see. Users do not see the distinguished name specified in the **DN** field.
- 5 In the **DN** field, enter the distinguished name of the searchbase as it appears in the directory. For example:  
`ou=searchbase, dc=domain, dc=example, dc=com`

---

**Note:** To include special characters such as a comma or backslash in a DN, you must escape the special character using another character. For instructions, see [“Using special characters in user names” on page 133](#).

---

- 6 To allow all users to search for recipients in the searchbase, select **Users can search**.  
Deselect this option in the rare circumstance when you do not want users to encrypt files for other users in the new searchbase. By default, this check box is selected.
- 7 Click **Apply**.
- 8 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
If the operation was successful, a success message appears.

# Modifying searchbases

Entrust PKI administrators with sufficient permissions can modify searchbases in Security Manager. You can modify the name of the searchbase, its distinguished name (DN), or whether all users can search for recipients in the searchbase.

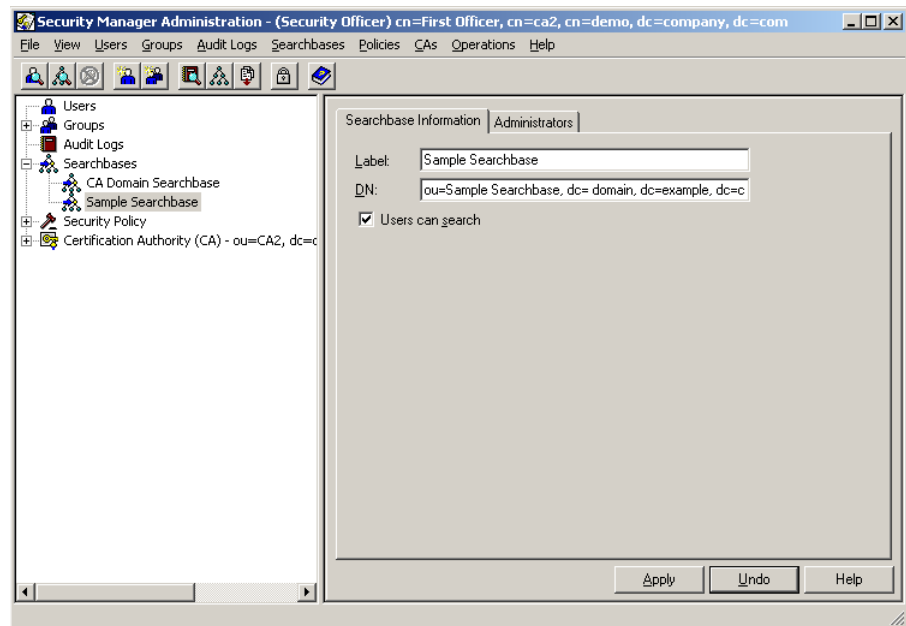
To change the DN of a searchbase, the new DN must already exist in the directory. If you use Active Directory, you must add the corresponding directory entry and assign permissions to that entry using your Active Directory administrative tools. For other supported directories, you can use the Directory Browser (see [“Adding entries to the directory” on page 74](#)).

You should change the DN of the searchbase only if you already changed the DN in the directory.

## To modify a searchbase

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 In the tree view, expand **Searchbases**.

A list of searchbases appears. The searchbases that appear depends on the list of searchbases that your own role allows you to access.



- 3 Select the searchbase that you want to modify.
- 4 To change the name of the searchbase, enter a new name into the **Label** field.

- 5** To change the distinguished name (DN) of the searchbase, enter the DN into the **DN** field.

To change the DN, the new DN must already exist in the directory. If you use Active Directory, you must add the corresponding directory entry and assign permissions to that entry using your Active Directory administrative tools. For other supported directories, you can use the Directory Browser (see [“Adding entries to the directory” on page 74](#)).

- 6** If you want to allow users to search for recipients in the searchbase, select **Users can search**. Otherwise, deselect the option.

Deselect this option in the rare circumstance when you do not want users to encrypt files for other users in the searchbase.

- 7** Click **Apply**.

- 8** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

# Deleting searchbases

Entrust PKI administrators with sufficient permissions can delete a searchbase. You might delete a searchbase for several reasons, including:

- When organizational units are used to distinguish separate locations (for example, the Ottawa [ou=Ottawa Office] and Washington offices [ou=Washington Office], and an office is closed or offices are merged).
- After two Certification Authorities (CAs) revoke cross-certification (users should not attempt to encrypt for users in the other CA). For more information about cross-certification, see [“Cross-certifying with other CAs” on page 465](#).

When you delete a searchbase, the associated directory entry is not automatically deleted. Users belonging to the searchbase are not deleted from the directory. Rather, users' association to the deleted searchbase are removed.

If you accidentally delete a searchbase, add it again using the same name, as described in [“Adding searchbases” on page 344](#). Then add the members to the restored searchbase.

## To delete a searchbase

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** In the tree view, expand **Searchbases**.  
A list of searchbases appears. The searchbases that appear depends on the list of searchbases that your own role allows you to access.
- 3** Select the searchbase that you want to delete.
- 4** Select **Searchbases > Selected Searchbase > Delete**.
- 5** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.





## Administering roles

Every user in Security Manager has a role. Roles control what operations users can perform, and which users and other system objects (such as searchbases and groups) the user can perform the operations on.

Security Officers (users with the predefined Security Officer role) have the most administrative capabilities, with permissions to administer all users and system objects. You can administer user roles to meet the specific needs of your organization.

This section describes how to view, create, and edit roles. This section contains the following topics:

- [“Predefined Security Manager user roles” on page 354](#)
- [“Viewing roles” on page 358](#)
- [“Creating roles” on page 361](#)
- [“Modifying roles” on page 363](#)
- [“Checking permission dependencies” on page 368](#)
- [“Deleting roles” on page 370](#)
- [“Permissions reference” on page 371](#)

# Predefined Security Manager user roles

Security Manager includes several predefined user roles that you can use in your PKI system, or use as a basis for creating custom roles (see [“Creating roles” on page 361](#)). Table 18 describes the predefined roles included with Security Manager.

**Table 18:** Predefined roles in Security Manager

Role	Description
Security Officer	<p>This role is for a few highly trusted people in your organization who use Security Manager Administration to administer sensitive operations. You need at least one Security Officer. Security Officers mainly set the security policy for your PKI and administer other Entrust PKI administrators.</p> <p>Security Officers use Security Manager Administration to perform tasks such as:</p> <ul style="list-style-type: none"><li>• configuring Security Manager so that it conforms to your organization’s security policies</li><li>• adding and deleting other Security Officers, Administrators, Auditors, and Directory Administrators</li><li>• cross-certifying with other CAs</li></ul> <p>You can modify this role by changing its name, the number of authorizations required for sensitive operations, and the policy certificate contents. You can also use this role as a basis for creating a custom role.</p>
Administrator	<p>This role is for any number of highly trusted people in your organization. For convenience, and depending on the size and nature of your user community, you may wish to have several people with this role distributed among your user community.</p> <p>Administrators with this role can use an administrative application (such as Security Manager Administration) to perform user-oriented tasks, such as</p> <ul style="list-style-type: none"><li>• activating, deactivating, and reactivating Entrust PKI users</li><li>• changing DNs (distinguished names) of Entrust PKI users</li><li>• revoking certificates</li><li>• recovering users (for an explanation of key recovery, see <a href="#">“Recovering user key pairs” on page 162</a>)</li></ul> <p>You can modify this role by changing its name, the number of authorizations required for sensitive operations, and the policy certificate contents. You can also use this role as a basis for creating a custom role.</p>

**Table 18:** Predefined roles in Security Manager (continued)

Role	Description
End User	<p>This role is for all non-administrators. End users have no administrative permissions.</p> <p>You can modify this role by changing its name and policy certificate contents. You can also use this role as a basis for creating a custom role.</p>
Directory Administrator	<p>This role is for any number of highly trusted people in your organization. Directory Administrators use the Directory Browser tool in Security Manager Administration to perform directory-related tasks, such as</p> <ul style="list-style-type: none"><li>• adding entries to and deleting them from the directory, either one at a time or in bulk</li><li>• adding attributes, such as email attributes, to directory entries so that Entrust PKI users can send and receive secure email</li><li>• changing user DNs in the directory, either one at a time or in bulk</li></ul> <p>You can modify this role by changing its name, the number of authorizations required for sensitive operations, and the policy certificate contents. You can use this role as a basis for creating a custom role.</p>
Auditor	<p>This role is for any number of highly trusted people in your organization. Auditors have a view-only role in Security Manager Administration—they can view but not modify audit logs, reports, the security policy, and user properties.</p> <p>You can modify this role by changing its name and the policy certificate contents. You can also use this role as a basis for creating a custom role.</p>
ASH Service	<p>The ASH Service profile uses this role. The ASH Service profile (ca.epf) is used by the Administration Service Handler (ASH) and the XML Administration Protocol (XAP) subsystems, and for moving and archiving users. Do not configure the ASH Service profile with the CA key pair, which is used to sign users' certificates.</p> <p>You can modify this role by changing its name, and the policy certificate contents. You can also use this role as a basis for creating a custom role.</p>
EAC Administrator	<p>This role is for CVCA administrators or DV administrators who log in to the CVCA Administration or DV Administration interfaces of Entrust Authority Administration Services. Administrators with this role can perform all tasks in the interface.</p> <p>This role is fully customizable.</p>

**Table 18:** Predefined roles in Security Manager (continued)

Role	Description
EAC Auditor	<p>This role is for CVCA administrators and DV administrators who log in to the CVCA Administration or DV Administration interfaces of Entrust Authority Administration Services. Administrators with this role can only view information in the interface.</p> <p>This role is fully customizable.</p>
EAC DV CKM Administrator	<p>This role is for the Automated DV Certificate Key Management Service of Entrust Authority Administration Services. It allows a Document Verifier to automatically request Document Verifier certificates from one or more CVCAs without intervention from an administrator.</p> <p>This role is fully customizable.</p>
EAC Self-Service	<p>This role is for the DV Web Service of Entrust Authority Administration Services. It allows a Document Verifier to exchange certificates with one or more Inspection Systems without administrator intervention.</p> <p>This role is fully customizable.</p>
Master List Signer Administrator	<p>This role is for Master List Signer administrators.</p> <p>You can modify this role by changing its name and the policy certificate contents. You can also use this role as a basis for creating a custom role.</p>
Self-Administration Server Administrator	<p>Entrust Authority Self-Administration Server requires a profile configured with this role to initialize the Admin Center. For more information about Self-Administration Server, see the Self-Administration Server documentation.</p> <p>This role is fully customizable.</p>
Server Login	<p>Entrust server products can use this role (for example, the User Management Service of Entrust Authority Administration Services requires a profile set up with this role). This role is a clone of the End User role, except that it uses the Server Login Policy user policy.</p> <p>You can modify this role by changing its name and the policy certificate contents. You can also use this role as a basis for creating a custom role.</p>
SPOC Administrator	<p>This role is for Single Point of Contact (SPOC) administrators.</p> <p>You can modify this role by changing its name and the policy certificate contents. You can also use this role as a basis for creating a custom role.</p>

**Table 18:** Predefined roles in Security Manager (continued)

Role	Description
SPOC Role	<p>This role is for SPOC Web Service profiles.</p> <p>You can modify this role by changing its name and the policy certificate contents. You can also use this role as a basis for creating a custom role.</p>
SPOC Self-Service Role	<p>This role is for the SPOC Domestic Web Service of Administration Services. It allows a Single Point of Contact (SPOC) to submit certificate requests from foreign SPOCs to the domestic CVCA.</p> <p>This role is fully customizable.</p>
User Administrator	<p>This role is intended for PKI administrators who use Administration Services. The User Administrator role is allowed to create and administer only those users whose certificate types are associated with user entities (as opposed to server entities).</p> <p>This role is fully customizable.</p>
User Reg Service (Admin Services)	<p>This role is for the User Registration Service in Administration Services. See the Administration Services documentation for details.</p> <p>This role is fully customizable.</p>

# Viewing roles

Entrust PKI administrators with sufficient permissions can view roles in Security Manager Administration. Entrust PKI administrators can also view a list of administrators that can administer a selected role. Entrust PKI administrators can only view the roles that their own role allows them to view. See [“Permissions reference” on page 371](#) for a list of all administrative permissions.

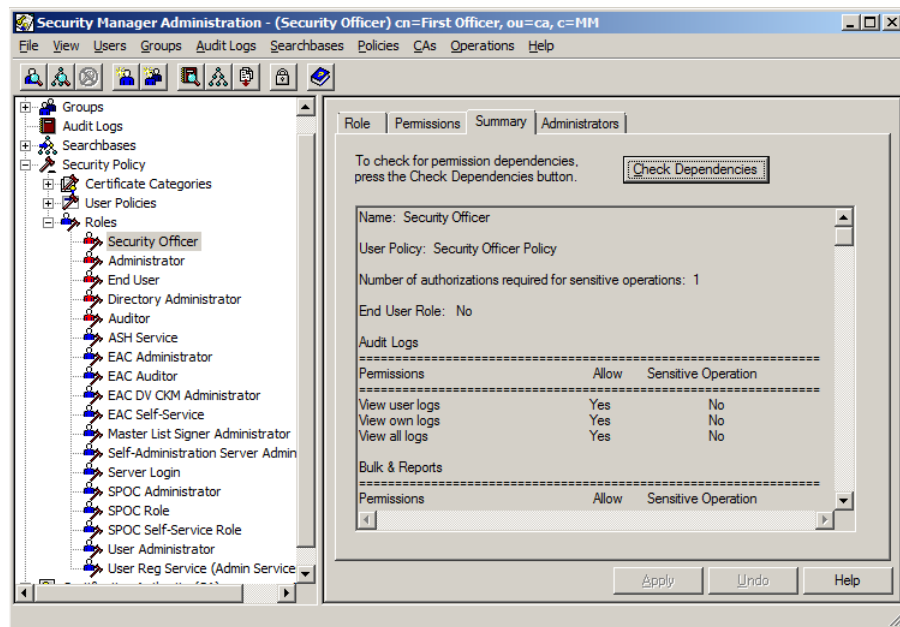
This section contains the following procedures:

- [“To view a role and its permissions” on page 358](#)
- [“To view a list of administrators that can administer a role” on page 359](#)

## To view a role and its permissions

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 In the tree view, expand **Security Policy > Roles**.

A list of roles appears. The roles that appear depends on the list of roles that your own role allows you to access.



Roles with red icons are read-only roles. Read-only roles always appear at the top of the list and in the order of Security Officer, Administrator, End User, Directory Administrator, and then Auditor, even if you renamed the role (see [“Modifying](#)

roles" on page 363). Therefore, a Security Officer role that was renamed to Senior Administrator still appears first in the list.

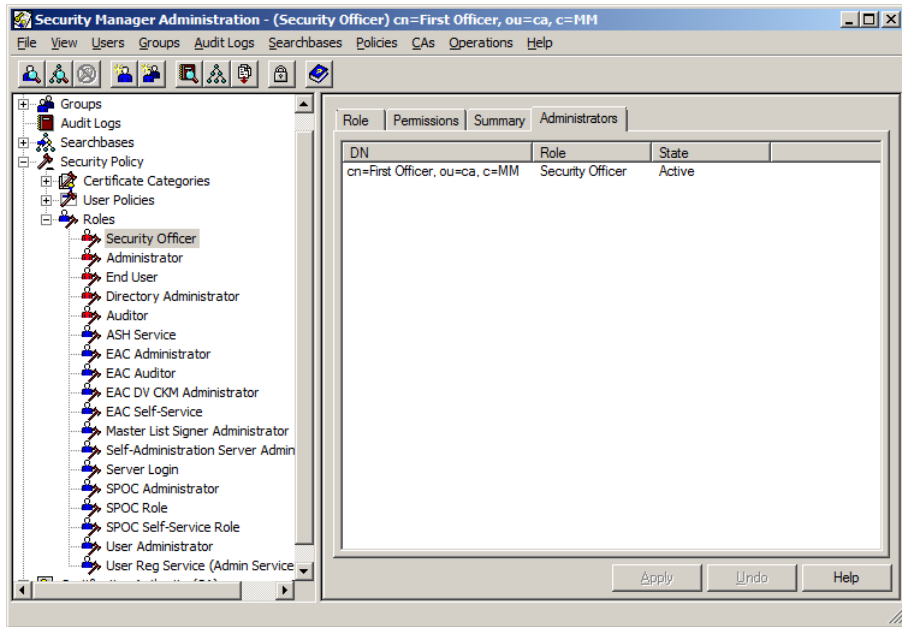
Roles with blue icons are custom roles. They appear alphabetical order beneath the roles with red icons. Security Manager includes several predefined custom roles.

For more information about the predefined roles, see ["Predefined Security Manager user roles" on page 354](#).

- 3** Select the role that you want to view. The role's property sheet appears in the right pane.
- 4** Click the **Summary** tab.  
The **Summary** property page displays a summary of the role.
- 5** (Optional.) If you want to save the summary for future reference, you can select and copy the summary into a text editor and then save it to a file.
- 6** (Optional.) To check the permission dependencies of the role, click **Check Dependencies**. For more information, see ["Checking permission dependencies" on page 368](#).

#### To view a list of administrators that can administer a role

- 1** Log in to Security Manager Administration (see ["Logging in to Security Manager Administration" on page 46](#)).
- 2** In the tree view, expand **Security Policy > Roles**.
- 3** Select the role that you want to view.
- 4** Click the **Administrators** tab to view the list of all Entrust PKI administrators who can administer the selected role.





# Creating roles

Every user in Security Manager has a role. Roles control what operations users can perform, and which users and other system objects (such as searchbases and groups) the user can perform the operations on.

Roles consist of:

- a name
- the number of authorizations required for sensitive authorizations
- a user policy

For information about user policies, see [“Administering user policies” on page 391](#).

- whether the role is an end user role
- permissions that control what operations the role can perform

For a list of all permissions, see [“Permissions reference” on page 371](#).

Besides the predefined roles provided with Security Manager (see [“Predefined Security Manager user roles” on page 354](#)), you can create new roles. You can create administrative roles or end user roles. Administrative roles allow users to administer users and other system objects in Security Manager. End user roles have no administrative permissions.

---

**Note:** If you create an end user role, you cannot later make that role an administrative role. If you create an administrative role, you cannot later make that role an end user role.

---

By narrowly defining the capabilities of roles you create, you can improve the security of your PKI. For example, you can create roles to administer specific groups of users.

Create administrative roles when you need Entrust PKI administrators with different permissions than the predefined roles offer. Typically you create new administrative roles to restrict the operations your Entrust PKI administrators can perform. For example, you may create roles to administer different searchbases, groups, or certificate types.

Create end user roles when you want end users to have different user policies. For example, you can create an end user role for roaming users, and an other end user role for desktop users.

To create a role, you must have sufficient permissions. By default, only Security Officers can create roles. To create a role, you can create a new role, or you can copy an existing role and then change the role's settings and permissions as required. Create a new role when you want a new role with few permissions, or when you want to minimize the chance of granting unintended permissions. Copy an existing role when you want a new role that is similar to an existing role.

This section contains the following procedures:

- [“To create a new role” on page 362](#)
- [“To create a role by copying an existing role” on page 362](#)

### To create a new role

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Select **Policies > Roles > New**.

A role with the name **<New Role>** and a blue icon appears in the tree view, and the new role's properties appear in the right pane.

---

**Note:** If you start to create a new role and change your mind at this point, then you can back out of the operation by selecting **Policies > Roles > Refresh Roles**.

---

- 3** Edit the new role. See [“Modifying roles” on page 363](#) for details.

### To create a role by copying an existing role

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** In the tree view, expand **Security Policy > Roles**.
- 3** Select the role that you want to copy and then select **Policies > Roles > Selected Role > Copy**.

A copy of the role appears at the bottom of the list of roles in the tree view, and the new role's properties appear in the right pane.

---

**Note:** If you start to create a new role and change your mind at this point, then you can back out of the operation by clicking **Undo**.

---

- 4** Customize the role's settings. For details, see [“Modifying roles” on page 363](#).

# Modifying roles

You can modify roles in Security Manager. Typically, you modify a role if you just created a new role or copied an existing role (see [“Creating roles” on page 361](#)). You can also modify any existing roles.

---

**Note:** You cannot modify the permissions of read-only roles (roles with a red icon), or the permissions of an end user role. You cannot change an administrative role into an end user role, and you cannot change an end user role into an administrative role. For more information about administrative roles and end user roles, see [“Creating roles” on page 361](#).

---

When you modify a role, you can change the name of the role, the user policy, the number of authorizations required for sensitive operations, and the role's permissions. If you just created a new role or copied an existing role (see [“Creating roles” on page 361](#)), you can also specify whether the role is an end user role.

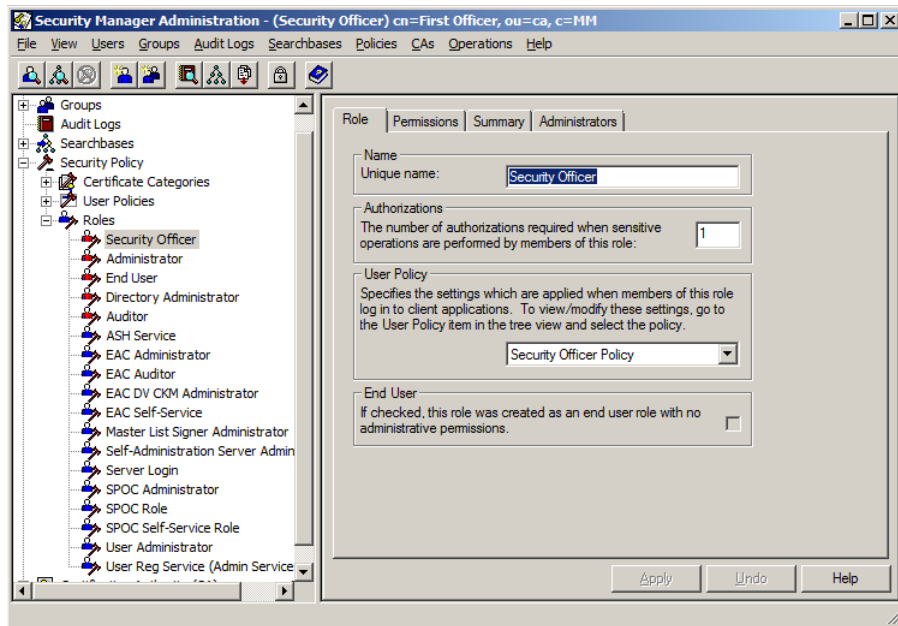
If you want members of a role to have different permissions, you must create a new role for each set of permissions that you want to give.

Entrust PKI administrators (users with administrative roles) cannot modify the permissions of a role if they do not already have permissions that are identical to the role they want to modify. For example, a user with the EAC Administrator role cannot modify the permissions of the User Administrator role. This is a security precaution. Another Entrust PKI administrator with corresponding permissions must make the required changes.

Changes to a role take effect for individual role members the next time the members log in to their Security Manager client application. Members that are currently logged in when you change their role continue to have their old settings until their certificates are updated.

## To change a role's settings

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 In the tree view, expand **Security Policy > Roles**.
- 3 Select the role that you want to modify.



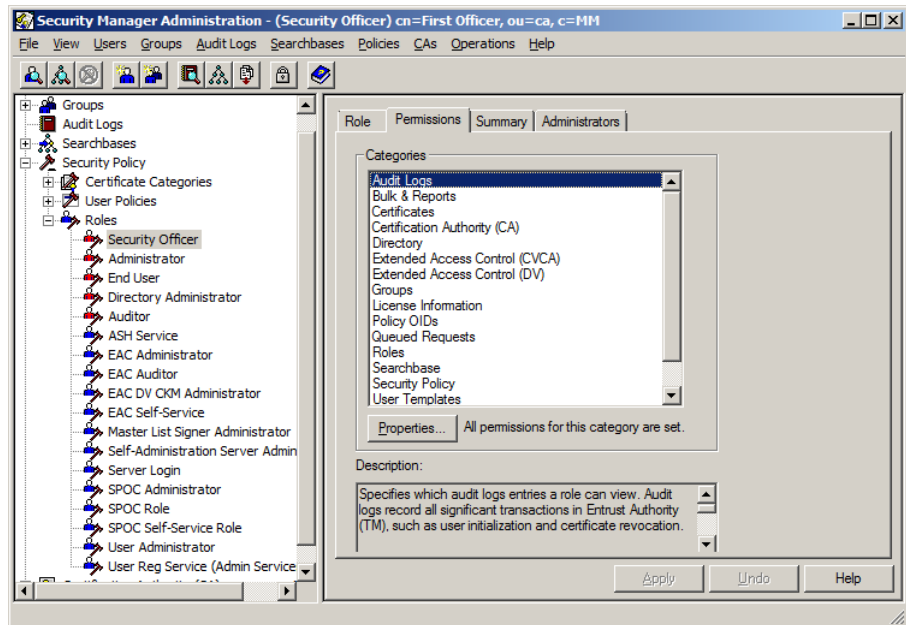
- 4 Click the **Role** tab.
- 5 To change the name of the role, enter a new name in the **Unique name** field.  
The name must contain at least one character that is not a tab or a space, and must not exceed 225 bytes (or 225 ASCII characters).
- 6 To change the number of authorizations required for sensitive operations, enter a new number in the **Authorizations** field.  
Do not enter a number that exceeds the total number of Entrust PKI administrators or you will not have enough Entrust PKI administrators to authorize operations. If you do not have enough administrators to authorize operations, a Security Officer must decrease the value. If you do not have enough Security Officers, a Master User must decrease the number of Security Officer authorizations, or set Security Officers to key-recovery mode in Security Manager Control Command Shell (see the *Security Manager Operations Guide* for details).
- 7 Under **User Policy**, select a user policy in the drop-down list.  
User policies define a role's default certificate settings. You can only assign one user policy to a role. For more information about user policies, see ["Administering user policies" on page 391](#).
- 8 If you are modifying a newly created role (see ["Creating roles" on page 361](#)), and you want to create the role as an end user role, select the **End User** check box.

---

**Note:** If you create the role as an end user role and apply the changes, the role can never have administrator permissions.

---

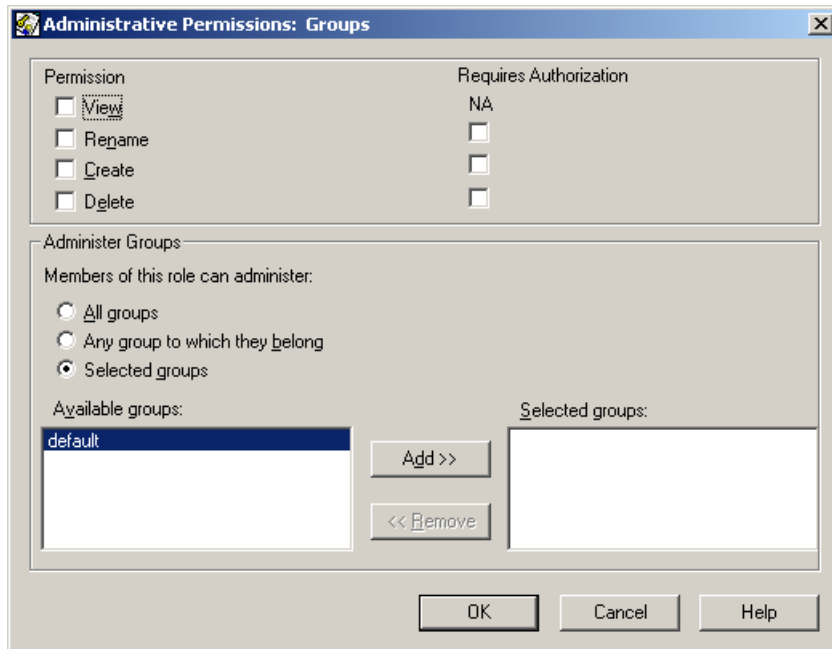
9 Click the **Permissions** tab.



10 Permissions specify which administrative tasks each role allows. For information about the available permissions, see [“Permissions reference” on page 371](#). To change a role’s permissions:

- Select a permissions category from the **Categories** list and then click **Properties**.

The **Administrative Permissions** dialog box for the permissions category appears. If you are modifying a predefined role (a role with a red icon) or an end user role, all options are disabled. You cannot modify the permissions for these roles.



**b** Configure permissions as follows:

- To allow the role to perform an operation, select the appropriate **Permission** check box.

When setting permissions, remember that you should combine certain permissions. For example, If you select a **Modify** permission (to modify roles, for example) you should also select the **View** permission (so the user can view the role, for example).

- To disallow an operation, deselect the **Permission** check box.
- To make an operation a sensitive operation that requires authorization from one or more Entrust PKI Administrators, select the **Requires Authorization** check box.

You cannot make all operations sensitive operations. Some operations, like the **View** permissions, cannot be sensitive operations. Operations that never require authorization have a **NA** (not applicable) instead of a check box.

- c** Some permission categories, such as Groups and Roles, allow you to specify whether the role can administer all objects of that type or a subset. For example, the Groups category allows you to specify whether an administrator with that role can administer all groups, only the groups to which the user belongs, or a specified list of groups.

Objects that the role can administer appear in the **Selected** list. For example, groups that the role can administer appear in the **Selected groups** list. Objects that the role cannot administer appear in the **Available** list. For example, groups that the role cannot administer appear in the **Available groups** list. You can move objects between lists by selecting the object and then clicking the **Add** and **Remove** buttons to move the object between lists. For example, to allow the role to administer a group, select the group from the **Available groups** list and then click **Add**.

**d** Click **OK**.

- 11** (Optional.) Click the Summary tab to view all the role's settings. Click **Check Dependencies** to check the role for permission dependencies. See ["Checking permission dependencies"](#) for more information about checking the role for permission dependencies.

If the **Show warnings about operations** setting in **File > Preferences > General** is turned on, permission dependencies are checked automatically whenever you change a role's permissions and click **Apply**.

- 12** Click **Apply** to save your changes or **Undo** to cancel. If you are modifying a role you just created (see ["Creating roles" on page 361](#)), clicking **Undo** deletes the role.

---

**Note:** It is faster to click **Apply** once, after making all your changes, rather than each time you change a category of permissions.

---

- 13** If prompted, authorize the operation. See ["Authorizing sensitive operations" on page 52](#).

If the operation was successful, a success message appears.

- 14** If you changed the number of authorizations required for a role, you must restart Security Manager Administration.

# Checking permission dependencies

Although you can set permissions independently of each other, you should set some permissions only when you set certain other permissions. For example, if you set the permission to modify roles, you should also set the permission to view roles, since you typically need to view a role before you can modify the role.

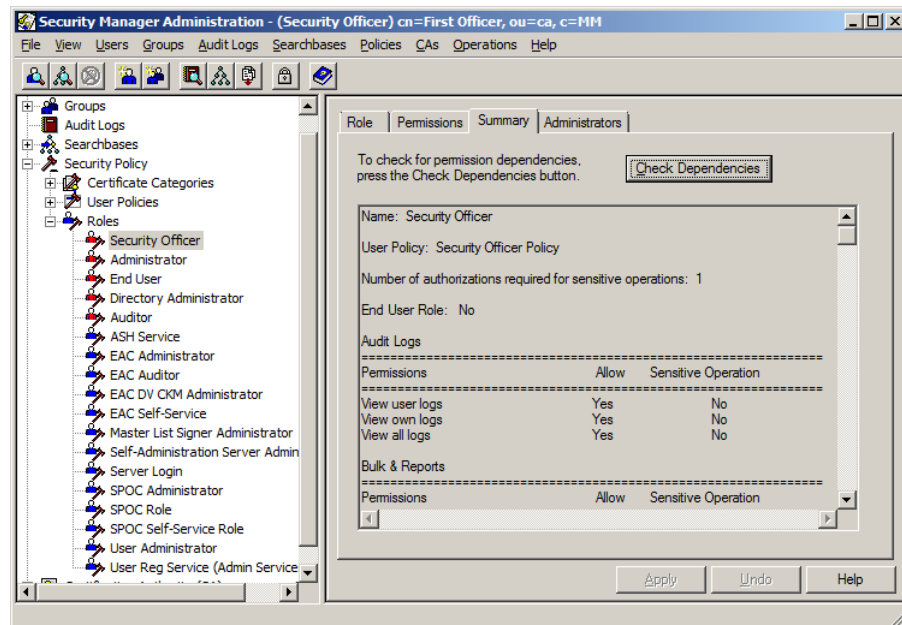
When you set permissions, you can check whether there are dependencies with other permissions. If there are dependencies, a dialog box tells you which additional permissions to set.

Note that when you check permissions for the Directory Administrator role, the message that appears recommends that you set other permissions. Ignore this message.

## To check permission dependencies

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 In the tree view, expand **Security Policy > Roles**.

A list of roles appears. The roles that appear depends on the list of roles that your own role allows you to access.



- 3 Select a role. The role's property sheet appears in the right pane.
- 4 Click the **Summary** tab.



The **Summary** property page displays a summary of the role.

**5** Click the **Check Dependencies** button.

Entrust checks the role's permissions for dependencies and displays the results in a dialog box.

Note that when you check permissions for the Directory Administrator role, the message that appears recommends that you set other permissions. Ignore this message.

**6** If required, add or remove permissions from the role (see "[Modifying roles](#)" on [page 363](#)) and then check dependencies again.

Ideally, you should have no dependencies. However, depending on the required capabilities of the role you are checking, one or more dependencies may be unavoidable.

# Deleting roles

If required, you can delete roles. You should only delete a role if your organization no longer requires that role. You cannot delete a role if that role is assigned to users. You cannot delete predefined roles (roles with a red icon). It is recommended that you do not delete the custom roles included with Security Manager. For a list of the default Security Manager roles, see [“Predefined Security Manager user roles” on page 354](#).

## To delete a role

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** In the tree view, expand **Security Policy > Roles**.
- 3** Select the role that you want to delete.
- 4** Select **Policies > Roles > Selected Role > Delete**.
- 5** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

# Permissions reference

Permission settings specify which administrative tasks each role allows. Most permissions allow members of a role to do specific administrative tasks, such as adding users to Entrust. Other permissions specify the scope of an operation, such as the list of groups that role members can administer.

When you create or modify roles, you can specify how many authorizations are required for sensitive operations (see [“Modifying roles” on page 363](#)). Most permissions allow you to specify whether the operations allowed by those permissions are sensitive operations that require authorization.

Note that some role permissions require authorization (**Requires Authorization** is selected), even though the role does not have permission to perform the operation. In these cases, the **Requires Authorization** setting is a recommended default if you copy the role and grant it the permission to the new role.

The following topics describe individual permission categories:

- [“Audit log permissions” on page 372](#)
- [“Bulk operations and reports permissions” on page 373](#)
- [“Certificate permissions” on page 373](#)
- [“Certification Authority \(CA\) permissions” on page 374](#)
- [“Directory permissions” on page 376](#)
- [“Extended Access Control \(CVCA\) permissions” on page 376](#)
- [“Extended Access Control \(DV\) permissions” on page 379](#)
- [“Group permissions” on page 382](#)
- [“License information permissions” on page 382](#)
- [“Policy OIDs permissions” on page 382](#)
- [“Queued requests permissions” on page 383](#)
- [“Role permissions” on page 384](#)
- [“Searchbase permissions” on page 384](#)
- [“Security policy permissions” on page 385](#)
- [“User template permissions” on page 386](#)
- [“User permissions” on page 386](#)

## Audit log permissions

These permissions specify which audit logs that role members can view. Audit logs record all significant transactions that occur in Security Manager, such as initializing users, revoking certificates, and so on. For more information about audit logs, see [“Working with audit logs” on page 609](#).

**Table 19:** Audit log permissions

Permission	Description
View user logs	<p>Controls whether role members can view user-related audit logs. To access a user-related audit, the role must also allow role members to administer the searchbase to which the user belongs (see <a href="#">“Searchbase permissions” on page 384</a>).</p> <p>User-related audits:</p> <ul style="list-style-type: none"><li>• have an audit code that falls within the range of X.509 audits (see the <i>Security Manager Operations Guide</i>)</li><li>• include a Target Name that specifies the distinguished name (DN) of a user</li></ul> <p>The Target Name identifies the user or entity to which the action reported by the audit was applied. In the Security Manager audit logs, the Target Name is identified by “Done to &gt;”.</p> <p>Some audits that could be considered as user-related audits are not actually classified as user-related audits, or may not include a DN in the Target Name. For example, audits related to attribute certificates are not classified as user-related audits. A role with only the <b>View user logs</b> permission cannot view these audits.</p> <p>Some audits that are classified of user-related audits may also apply to other types of entities, such as subordinate CAs. If the audit includes a Target Name which belongs to a searchbase that the role can administer, roles with the <b>View user logs</b> permission can view these audits.</p>
View own logs	Controls whether role members can view the audit logs that they generate in Security Manager.
View all logs	Controls whether role members can view all audit logs generated in Security Manager.

# Bulk operations and reports permissions

These permissions specify whether role members can perform bulk operations or create reports. Bulk operations allow users to perform multiple operations at once, such as adding multiple users. Reports can list such things as all added users, all users that are set up for key recovery, and so on.

For more information about bulk operations, see [“Performing bulk operations” on page 275](#). For more information about reports, see [“Working with audit logs and creating reports” on page 607](#).

**Table 20:** Bulk operations and report permissions

Permission	Description
Process bulk files	Controls whether role members can use the Bulk Console to perform bulk operations.
Create reports	Controls whether role members can create reports on the Security Manager database.

# Certificate permissions

These permissions specify which certificate categories and types of certificates that role members can view and administer. For more information about certificates, see [“Customizing certificates” on page 525](#). Certificate permissions are divided into the following categories:

- [“Certificate categories permissions” on page 373](#)
- [“Certificate types permissions” on page 374](#)

## Certificate categories permissions

These permissions specify which certificate categories that role members can view and administer. Role members can also view Security Manager license information for that certificate category, provided the role also allows members to administer license information (see [“License information permissions” on page 382](#)).

For example, if you can administer the Enterprise certificate category, you can assign Enterprise certificates to users and view the Enterprise license information.

For more information about certificates, see [“Customizing certificates” on page 525](#).

**Table 21:** Certificate categories permissions

Permission	Description
Administer Categories	<p>Controls which certificate categories that role members can view and administer. Role members can administer:</p> <ul style="list-style-type: none"><li>• <b>All categories</b> (role members can administer all certificate categories)</li><li>• <b>Selected categories</b> (role members can administer a specified list of certificate categories)</li></ul> <p>If a role contains no certificate categories to administer, role members cannot view or modify any user entries, or view any security policy settings.</p>

### Certificate types permissions

These permissions specify which types of certificates that role members can view and administer. For example, if you have permission to administer the Default Enterprise certificate type, you can assign the Default Enterprise certificate type to users.

For more information about certificates, see [“Customizing certificates” on page 525](#).

**Table 22:** Certificate types permissions

Permission	Description
Administer Types	<p>Controls which certificate types users can view and administer. Users can administer:</p> <ul style="list-style-type: none"><li>• <b>All types</b> (users can administer all certificate types)</li><li>• <b>Selected types</b> (users can administer a specified list of certificate types)</li></ul> <p>If a role contains no certificate types to administer, role members cannot view or modify any user entries, or view any security policy settings.</p>

### Certification Authority (CA) permissions

These permissions specify the operations that role members can perform on Certification Authorities (CAs). Certification Authority permissions are divided into the following categories:

- [“CA permissions” on page 375](#)
- [“Cross-certified CA permissions” on page 375](#)

- [“Subordinate CA permissions” on page 376](#)

## CA permissions

These permissions specify the operations that role members can perform on their own Certification Authority.

**Table 23:** CA permissions

Permission	Description
View CA certificates	Controls whether role members can view CA certificates.
Update CA signing keys	Controls whether role members can update a CA's signing keys.
Revoke CA keys	Controls whether role members can revoke a CA's keys. Only Master Users can revoke CA keys. See the <i>Security Manager Operations Guide</i> for details.
View list of imported CAs	Controls whether role members can view the list of imported CAs.
Import/export CA public keys	Controls whether role members can import and export a CA's public keys.

## Cross-certified CA permissions

These permissions specify the operations that role members can perform on cross-certified CAs.

**Table 24:** Cross-certified CA permissions

Permission	Description
View	Controls whether role members can view cross-certified CAs.
Initiate	Controls whether role members can begin to cross-certify with another CA.
Revoke	Controls whether role members can revoke cross-certified CAs.
Complete	Controls whether role members can finish cross-certifying with another CA.

## Subordinate CA permissions

These permissions specify the operations that role members can perform on subordinate CAs.

**Table 25:** Subordinate CA permissions

Permission	Description
View	Controls whether role members can view subordinate CAs.
Add	Controls whether role members can add subordinate CAs.
Revoke	Controls whether role members can revoke subordinate CAs.

## Directory permissions

These permissions specify whether role members can bind to the directory or change the directory password. They also specify whether you can use the Directory Browser tool to view or administer directory entries. For more information about the Directory Browser, see [“Using the Directory Browser” on page 69](#).

**Table 26:** Directory permissions

Permission	Description
Bind to Directory	Controls whether role members can bind to the Security Manager directory.
Change Directory password	Controls whether role members can change the password Security Manager Administration uses to access the Security Manager directory.
View entries	Controls whether role members can use the Directory Browser to view entries in the Security Manager directory.
Create, delete, modify entries	Controls whether role members can use the Directory Browser to create, delete, or modify entries in the Security Manager directory.

## Extended Access Control (CVCA) permissions

These permissions specify the administrative operations that role members can perform on a Country Verifying Certification Authority (CVCA). CVCA permissions are divided into the following categories:

- [“Anchor CVCA permissions” on page 377](#)
- [“DV permissions” on page 377](#)
- [“Foreign CVCA permissions” on page 378](#)



## Anchor CVCA permissions

These permissions specify the operations that role members can perform on the domestic (anchor) CVCA.

**Table 27:** Anchor CVCA permissions

Permission	Description
View Anchor Policy	Controls whether role members can view the CVCA policy and Document Verifier policy.
Modify Anchor Policy	Controls whether role members can modify the CVCA policy and Document Verifier policy. Requires the <b>View Anchor Policy</b> permission.
View Anchor Cert	Controls whether role members can view the domestic (anchor) CVCA's certificates.

## DV permissions

These permissions specify the operations that role members can perform on trusted Document Verifiers.

**Table 28:** Document Verifier permissions

Permission	Description
View DV	Controls whether role members can view Document Verifiers added to the CVCA.
Add DV	Controls whether role members can add Document Verifiers to the CVCA. Requires the <b>View DV</b> permission.
Modify DV	Controls whether role members can modify Document Verifiers. Requires the <b>View DV</b> permission.
Disable DV	Controls whether role members can disable Document Verifiers. Requires the <b>View DV</b> permission.
Enable DV	Controls whether role members can enable Document Verifiers. Requires the <b>View DV</b> permission.
Delete DV	Controls whether role members can delete Document Verifiers from the CVCA. Requires the <b>View DV</b> permission.
View DV Cert	Controls whether role members can view Document Verifier certificates that were issued by the CVCA.

**Table 28:** Document Verifier permissions (continued)

Permission	Description
Process Auth DV Certreq	Controls whether role members can process authenticated DV certificate requests. Requires the <b>View DV Cert</b> permission.
Process Unauth Certreq	Controls whether role members can process unauthenticated DV certificate requests. Requires the <b>Process Auth Certreq</b> permission.
Process Expired DV Certreq	Controls whether role members can process DV certificate requests authenticated by an expired DV certificate. Requires the <b>Process Auth Certreq</b> permission.
Preview DV Certreq	Controls whether role members can preview DV certificate requests.
Countersign Auth DV Certreq	Controls whether role members can countersign authenticated DV certificate requests. Requires the <b>Preview DV Certreq</b> permission.
Countersign Unauth DV Certreq	Controls whether role members can countersign unauthenticated DV certificate requests. Requires the <b>Countersign Auth DV Certreq</b> permission.

### Foreign CVCA permissions

These permissions specify the operations that role members can perform on trusted foreign CVCA.

**Table 29:** Foreign CVCA permissions

Permission	Description
View FCVCA	Controls whether role members can view foreign CVCA.
Add FCVCA	Controls whether role members can add foreign CVCA to the domestic (anchor) CVCA. Requires the <b>Add FCVCA</b> permission.
Modify FCVCA	Controls whether role members can modify foreign CVCA. Requires the <b>Add FCVCA</b> permission.
Disable FCVCA	Controls whether role members can disable foreign CVCA. Requires the <b>Add FCVCA</b> permission.
Enable FCVCA	Controls whether role members can enable foreign CVCA. Requires the <b>Add FCVCA</b> permission.
Delete FCVCA	Controls whether role members can delete foreign CVCA. Requires the <b>Add FCVCA</b> permission.

**Table 29:** Foreign CVCA permissions (continued)

Permission	Description
View FCVCA Cert	Controls whether role members can view foreign CVCA certificates.
Import FCVCA Link Cert	Controls whether role members can import foreign CVCA link certificates. Requires the <b>View FCVCA Cert</b> permission.
Import FCVCA Root Cert	Controls whether role members can import foreign CVCA root certificates. Requires the <b>Import FCVCA Link Cert</b> permission.

## Extended Access Control (DV) permissions

These permissions specify the administrative operations that role members can perform on a Document Verifier (DV). Document Verifier permissions are divided into the following categories:

- [“Anchor DV permissions” on page 379](#)
- [“CVCA permissions” on page 380](#)
- [“Inspection System permissions” on page 381](#)

### Anchor DV permissions

These permissions specify the operations that role members can perform on the anchor Document Verifier.

**Table 30:** Anchor Document Verifier permissions

Permission	Description
View Anchor Policy	Controls whether role members can view the CVCA, Document Verifier, and Inspection System policy.
Modify Anchor Policy	Controls whether role members can modify the CVCA, Document Verifier, and Inspection System policy. Requires the <b>View Anchor Policy</b> permission.
View Anchor Cert	Controls whether role members can view Document Verifier certificates issued by CVCAs.
Finish Anchor Certreq	Controls whether role members can finish processing a certificate request by importing the requested Document Verifier certificate. Requires the <b>View Anchor Cert</b> permission.
View Anchor Certreq	Controls whether role members can view DV certificate requests.

**Table 30:** Anchor Document Verifier permissions (continued)

Permission	Description
Create Anchor Certreq	Controls whether role members can create DV certificate requests. Requires the <b>View Anchor Certreq</b> permission.
Cancel Anchor Certreq	Controls whether role members can cancel DV certificate requests. Requires the <b>View Anchor Certreq</b> permission.
View Anchor Cert Chain	Controls whether role members can view certificate chains (from the CVCA to Document Verifier).

### CVCA permissions

These permissions specify the operations that role members can perform on trusted Country Verifying Certification Authorities (CVCAs).

**Table 31:** CVCA permissions

Permission	Description
View CVCA	Controls whether role members can view CVCAs added to the Document Verifier.
Add CVCA	Controls whether role members can add CVCAs. Requires the View CVCA permission.
Modify CVCA	Controls whether role members can modify CVCAs. Requires the View CVCA permission.
Disable CVCA	Controls whether role members can disable CVCAs. Requires the View CVCA permission.
Enable CVCA	Controls whether role members can enable CVCAs. Requires the View CVCA permission.
Delete CVCA	Controls whether role members can delete CVCAs. Requires the View CVCA permission.
View CVCA Cert	Controls whether role members can view CVCA certificates.
Import CVCA Link Cert	Controls whether role members can import CVCA link certificates. Requires the <b>View CVCA Cert</b> permission.
Import CVCA Root Cert	Controls whether role members can import CVCA root certificates. Requires the <b>Import CVCA Link Cert</b> permission.

## Inspection System permissions

These permissions specify the operations that role members can perform on trusted Inspection Systems.

**Table 32:** Inspection System permissions

Permission	Description
View IS	Controls whether role members can view Inspection Systems added to the Document Verifier.
Add IS	Controls whether role members can add Inspection Systems. Requires the <b>View IS</b> permission.
Modify IS	Controls whether role members can modify Inspection Systems. Requires the <b>View IS</b> permission.
Disable IS	Controls whether role members can disable Inspection Systems. Requires the <b>View IS</b> permission.
Enable IS	Controls whether role members can enable Inspection Systems. Requires the <b>View IS</b> permission.
Delete IS	Controls whether role members can delete Inspection Systems. Requires the <b>View IS</b> permission.
View IS Cert	Controls whether role members can view Inspection System certificates.
Process Auth IS Certreq	Controls whether role members can process authenticated Inspection System certificate requests. Requires the <b>View IS Cert</b> permission.
Process Unauth IS Certreq	Controls whether role members can process unauthenticated Inspection System certificate requests. Requires the <b>Process Auth IS Certreq</b> permission.
Process Expired IS Certreq	Controls whether role members can process expired Inspection System certificate requests. Requires the <b>Process Auth IS Certreq</b> permission.
View IS Cert Chain	Controls whether role members can view certificate chains (from the CVCA to the Inspection System).
Preview IS Certreq	Controls whether role members can preview Inspection System certificate requests.

## Group permissions

These permissions specify the administrative operations that role members can perform on groups, and specifies which groups that role members can administer. For more information about groups, see [“Administering groups” on page 329](#).

**Table 33:** Group permissions

Permission	Description
View	Controls whether role members can view groups.
Rename	Controls whether role members can rename groups.
Create	Controls whether role members can create new groups.
Delete	Controls whether role members can delete groups.
Administer Groups	Specifies which groups that role members can administer. Role members can administer: <ul style="list-style-type: none"><li>• <b>All groups</b> (role members can administer all groups)</li><li>• <b>Any group to which the they belong</b> (role members can administer only the groups to which they belong)</li><li>• <b>Selected groups</b> (role members can administer a specified list of groups)</li></ul>

## License information permissions

These permissions specify whether role members can view or modify Security Manager license information (the serial number and user limit) for their allowed certificate categories (see [“Certificate permissions” on page 373](#)). For more information about viewing and changing the license information, see [“Configuring the Security Manager license information” on page 55](#).

**Table 34:** License Information permissions

Permission	Description
View	Controls whether role members can view license information.
Modify	Controls whether role members can change license information.

## Policy OIDs permissions

These permissions specify which encryption and verification policy object identifiers (OIDs) you can assign to roles and users, and whether you can create or delete OIDs.

For information about configuring OIDs, see [“Configuring the encryption and verification OIDs” on page 90](#). For conceptual information about OIDs, see the *Security Manager Directory Configuration Guide*.

**Table 35:** Policy OIDs permissions

Permission	Description
Administer OIDs	Specifies which OIDs that role members can administer. Role members can administer: <ul style="list-style-type: none"><li>• <b>All OIDs</b> (role members can administer all OIDs)</li><li>• <b>Selected OIDs</b> (role members can administer a specified list of OIDs)</li></ul>

### Queued requests permissions

Each role contains a setting that specifies how many authorizations are required for sensitive operations. Most role permissions allow you to specify whether the operations allowed by those permissions are sensitive operations that require authorization.

When you create or modify roles, you can specify how many authorizations are required for sensitive operations, and which permissions require authorization. See [“Modifying roles” on page 363](#) for more information.

When an Entrust PKI administrator performs a sensitive operation that requires more than one authorization, other Entrust PKI administrators must approve the operation. Some Security Manager client applications, such as Administration Services, queue the requests for authorization, removing the need for other administrators to approve the operation immediately. Queued requests allows administrators to continue administering Security Manager users and data without interruption and allows administrators to view, cancel, and approve or reject the requests as required.

These permissions specify whether role members can view, modify, create, remove, or cancel queued administrative requests, and whether role members can cancel an Entrust PKI administrator’s authorization of a queued request.

**Table 36:** Queued Requests permissions

Permission	Description
View Queued Requests	Controls whether role members can view queued requests.
Modify Queued Requests	Controls whether role members can modify queued requests.
Create Queued Requests	Controls whether role members can create queued requests.

**Table 36:** Queued Requests permissions (continued)

Permission	Description
Delete Queued Requests	Controls whether role members can delete queued requests.
Cancel Queued Requests	Controls whether role members can cancel queued requests.
Cancel Request Authorization	Controls whether role members can cancel authorized requests.
Approve Request Authorization	Controls whether role members can approve queued requests.

## Role permissions

These permissions specify which administrative operations that role members can perform on roles, and which roles that role members can administer.

**Table 37:** Roles permissions

Permission	Description
View	Controls whether role members can view roles.
Modify	Controls whether role members can modify roles.
Create	Controls whether role members can create new roles.
Delete	Controls whether role members can delete roles.
Administer Roles	Specifies which roles that role members can administer. Role members can administer: <ul style="list-style-type: none"> <li>• <b>All roles</b> (role members can administer all roles)</li> <li>• <b>Selected roles</b> (role members can administer a specified list of roles)</li> </ul>

## Searchbase permissions

These permissions specify which administrative operations that role members can perform on searchbases, and which searchbases that role members can administer. Since searchbases map to the structure of your Directory Information Tree (DIT), you



can restrict a role to administering only users who appear in a certain part of the directory.

**Table 38:** Searchbase permissions

Permission	Description
View	Controls whether role members can view searchbases.
Modify	Controls whether role members can modify searchbases.
Create	Controls whether role members can add searchbases to Security Manager.
Delete	Controls whether role members can delete searchbases.
Administer Searchbases	Specifies which searchbases that role members can administer. Role members can administer: <ul style="list-style-type: none"><li>• <b>All searchbases</b> (role members can administer all searchbases)</li><li>• <b>Selected searchbases</b> (role members can administer a specified list of searchbases)</li></ul>

## Security policy permissions

These permissions specify which parts of the security policy that role members can administer. These settings govern how members of your organization use Security Manager to achieve security objectives.

**Table 39:** Security Policy permissions

Permission	Description
View Security Policy	Controls whether role members can view the security policy.
Modify Security Policy	Controls whether role members can modify the security policy.
Export Certificate Specification	Controls whether role members can export certificate specification files ( <code>master.certspec</code> ).
Import Certificate Specification	Controls whether role members can import certificate specification files ( <code>master.certspec</code> ).
Export User Templates	Controls whether role members can export user templates.
Import User Templates	Controls whether role members can import user templates.
Force CRLs	Controls whether role members can issue certificate revocation lists (CRLs) after revoking a user's certificates.

**Table 39:** Security Policy permissions (continued)

Permission	Description
View CRLs	Controls whether role members can view CRLs.
View User Policy	Controls whether role members can view user policies.
Modify User Policy	Controls whether role members can modify user policies.
Create User Policy	Controls whether role members can create user policies.
Delete User Policy	Controls whether role members can delete user policies.

## User template permissions

These permissions specify which user templates that role members can select to create new Security Manager users. User templates define user types by specifying what fields appear in the **New User** dialog box and what information those fields require. Two default user templates are Person and Web Server.

If you assign an empty list of user templates to the role, role members cannot create users in Security Manager Administration except through bulk operations, provided they have sufficient permissions to perform bulk operations (see [“Bulk operations and reports permissions” on page 373](#)).

For more information about choosing a user template when creating a user, see [“Creating new users” on page 146](#). For information about creating user templates, see [“Modifying the user template file and user types” on page 447](#).

**Table 40:** User Templates permissions

Permission	Description
Administer Templates	Specifies which user templates that role members can administer. Role members can administer: <ul style="list-style-type: none"><li>• <b>All templates</b> (role members can administer all user templates)</li><li>• <b>Selected templates</b> (role members can administer a specified list of templates)</li></ul>

## User permissions

These permissions specify the administrative operations that role members can perform on users. User permissions are divided into the following categories:

- [“General user permissions” on page 387](#)
- [“Advanced user permissions” on page 387](#)

- [“Other user permissions” on page 388](#)

## General user permissions

General user permissions govern common user administration tasks like adding users, changing user DNs, and updating user key pairs. These tasks have few security implications and need not be the kinds of things that only Security Officers can do.

**Table 41:** General user permissions

Permission	Description
View	Controls whether role members can view users.
Add	Controls whether role members can add users to Security Manager.
Reactivate	Controls whether role members can activate users who were deactivated.
Deactivate/Remove	Controls whether role members can deactivate users or remove them from Security Manager.
Change DN	Controls whether role members can change a user's distinguished name (DN).
Modify properties	Controls whether role members can modify a user's properties.
Revoke certificates	Controls whether role members can revoke a user's certificates.
Update key pairs	Controls whether role members can update a user's key pairs.
Set for key recovery	Controls whether role members can set users for key recovery.
Cancel key recovery	Controls whether role members can cancel a user's key recovery state.
Modify key update options	Controls whether role members can modify a user's key update options.
View activation codes	Controls whether role members can view a user's activation codes.
Reissue activation codes	Controls whether role members can reissue activation codes to users.

## Advanced user permissions

Advanced user permissions govern tasks that have greater security implications than the general tasks. For example, advanced tasks include changing a user's role and

importing users from another CA. These tasks are best suited for highly-trusted people in your organization, like Security Officers.

**Table 42:** Advanced user permissions

Permission	Description
Modify OIDS	Controls whether role members can modify the set of default object identifiers (OIDs) that users receive.
Change user's role	Controls whether role members can change a user's role.
Modify group membership	Controls whether role members can modify a user's group membership.
Import new user	Controls whether role members can import users from another CA.
Export to another CA	Controls whether role members can export users to another CA.
Archive users	Controls whether role members can archive users.
View archived users	Controls whether role members can view archived users.
Retrieve archived users	Controls whether role members can recover archived users.
Restore information to the Directory	Controls whether role members can restore a user's information to the Security Manager directory.
Perform PKIX requests	Controls whether role members can submit PKIX requests on behalf of other users or devices.
Create user profile	Controls whether role members can create digital IDs for users.
Recover user profile	Controls whether role members can recover user digital IDs.
Convert Protocol Version	Controls whether role members can convert a V2 user (a user with a V2 digital ID) to a V1 user (a user with a V1 digital ID).

## Other user permissions

Other user permissions govern miscellaneous user administration tasks such as those related to attribute certificates and registration passwords.

**Table 43:** Other user permissions

Permission	Description
View Attribute Certificate	Controls whether role members can view attribute certificates.

**Table 43:** Other user permissions (continued)

Permission	Description
Modify Attribute Certificate	Controls whether role members can modify attribute certificates.
Create Attribute Certificate	Controls whether role members can create attribute certificates.
Delete Attribute Certificate	Controls whether role members can delete attribute certificates.
View Registration Password	Controls whether role members can view user registration passwords.
Modify Registration Password	Controls whether role members can modify user registration passwords.
Validate Registration Password	Controls whether role members can validate user registration passwords.
Notify Client	Controls whether role members can notify Security Manager client applications of an event that affects the client's users. See <a href="#">"Notifying client applications" on page 242</a> for more information.
Modify Directory Properties	Controls whether role members can modify a entries in the Security Manager directory.
Symmetric Key Access	Controls whether role members can access user symmetric keys.



## Administering user policies

User policies (also called policy certificates) are divided into client policies and certificate definition policies. Client policies define the certificates associated with roles. Certificate definition policies define the certificates associated with user types. Security Manager includes several default user policies that you can use in your PKI system. You can create, modify, and delete user policies as required by your organization.

This chapter includes the following topics:

- [“User policy overview” on page 392](#)
- [“Predefined user policies” on page 393](#)
- [“Viewing user policies” on page 395](#)
- [“Viewing policy certificates” on page 397](#)
- [“Modifying user policies” on page 402](#)
- [“Creating user policies” on page 398](#)
- [“Mapping policy certificates to certificate definitions” on page 404](#)
- [“Deleting user policies” on page 405](#)
- [“Importing and exporting user policies” on page 406](#)
- [“Client policy attributes reference” on page 407](#)
- [“Certificate definition policy attributes reference” on page 432](#)

# User policy overview

User policies (also called policy certificates) are divided into client policies and certificate definition policies.

Client policies define the certificates associated with roles, so that the policy attributes you define apply to every user with the associated role. For example, when you define attributes for the End User role, every user with the End User role is governed by those policy attributes.

Certificate definition policies define the certificates associated with user types, so that the policy attributes you define apply to every user with the associated certificate. For example, when you define attributes for the encryption certificate definition, every user with the encryption certificate is governed by those policies.

Most users receive both types of user policies. They receive the certificate definition policy associated with each of their certificate definitions, and the client policy associated with their user role. Typically, their Security Manager client application takes information from both types of policies, and arbitrates between them if necessary. For example, the client application may take the certificate generation attributes from the certificate definition policy, ignoring any corresponding settings in the client policy, and take the password attributes from the client policy.

In circumstances where there the two policies have conflicting attributes (for example, the number of key pairs is different in the two user policies), the client application determines the priority. For more information, see your documentation for your client application.

Security Manager includes several default user policies that you can use in your PKI system (see [“Predefined user policies” on page 393](#)). You can create, modify, and delete user policies as required by your organization.



# Predefined user policies

Security Manager includes several predefined user policies that you can use in your PKI system, or use as a basis for creating custom user policies (see [“Creating user policies” on page 398](#)). Table 44 describes the default user policies included with Security Manager.

**Table 44:** Predefined Security Manager user policies

User Policy	Description
Administrator Policy	Client policy that defines a set of policies for Entrust PKI administrators.
ASH Policy	Client policy that defines a set of policies for the CA user.
Document Signer Policy	Certificate definition policy that defines a set of policies for Document Signer certificates.
Dual Usage No Key Backup Policy	Certification definition policy that defines a set of policies for dual-usage certificates with no key backup.
Dual Usage Policy	Certification definition policy that defines a set of policies for dual-usage certificates.
EFS Policy	Certification definition policy that defines a set of policies for EFS certificates.
Encryption Policy	Certification definition policy that defines a set of policies for encryption certificates.
Encryption_p10 Policy	Certification definition policy that defines a set of policies for PKCS#10 encryption certificates.
End User Policy	Client policy that defines a set of policies for all end users (non-administrative users).
Enterprise Domain Controller Policy	Certificate definition policy that defines a set of policies for Enterprise domain controller certificates.
Enterprise Machine Policy	Certificate definition policy that defines a set of policies for Enterprise computer digital IDs.
Master List Signer Administrator Policy	Client policy that defines a set of policies for Master List Signer administrators.
Master List Signer Policy	Certificate definition policy that defines a set of policies for Master List Signer certificates.

**Table 44:** Predefined Security Manager user policies (continued)

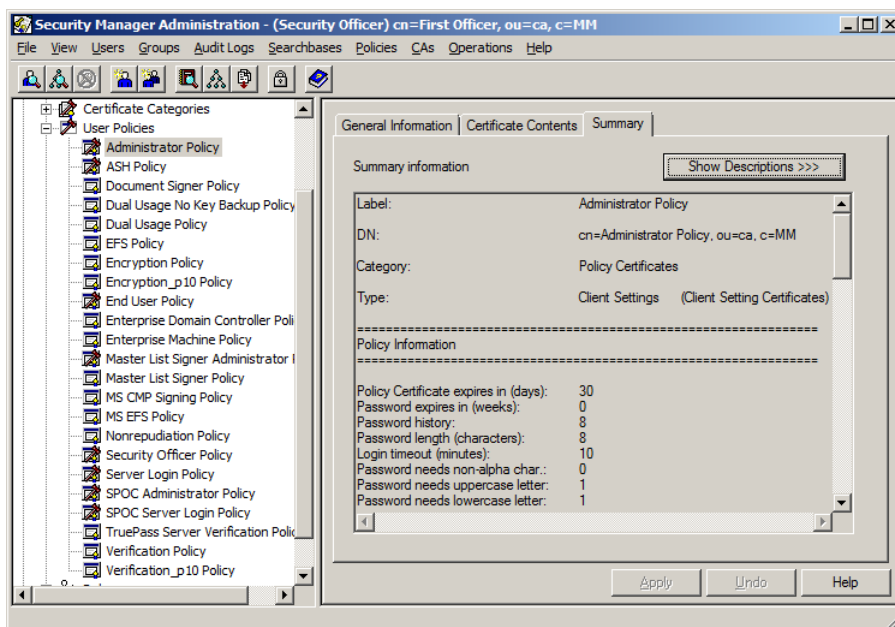
User Policy	Description
MS CMP Signing Policy	Certification definition policy that defines a set of policies for Microsoft CMP signing certificates.
MS EFS Policy	Certification definition policy that defines a set of policies for Microsoft EFS certificates.
Nonrepudiation Policy	Certification definition policy that defines a set of policies for nonrepudiation certificates.
Security Officer Policy	Client policy that defines a set of policies for Security Officers.
Server Login Policy	Client policy that defines a set of policies for Security Manager clients to log in to the Security Manager service. Not all Security Manager clients use this user policy. See the documentation for your Security Manager client to see if it uses this user policy.
SPOC Administrator Policy	Client policy that defines a set of policies for Single Point of Contact (SPOC) administrators.
SPOC Server Login Policy	Client policy that defines a set of policies for the SPOC Web Service in Entrust Authority Administration Services. The SPOC Web Service uses this policy to log in to the Security Manager service.
TruePass Server Verification Policy	Certification definition policy that defines a set of policies for Entrust. TruePass server verification certificates.
Verification Policy	Certification definition policy that defines a set of policies for verification certificates.
Verification_p10 Policy	Certification definition policy that defines a set of policies for PKCS#10 verification certificates.

# Viewing user policies

Entrust PKI administrators with sufficient permissions can view user policies in Security Manager Administration. When you view a user policy, you can copy a record of the policy settings to a file for reference.

## To view a user policy

- 1 Log in to Security Manager Administration as a Security Officer (see [“Logging in to Security Manager Administration”](#) on page 46).
- 2 In the tree view, expand **Security Policy > User Policies**.



- 3 Select the user policy that you want to view.
- 4 Click the **Summary** tab.

This property page shows all the settings for the user policy you selected, as well as the policy name, DN, category, and policy type.

The settings shown on this page are the current, applied settings. If you have changed any settings but not yet applied them, the settings display with their old values.

The values for Boolean (on/off) settings are shown as 1 for on, or 0 for off.

- 5 To include the descriptions of the policy settings, click **Show Descriptions**. (For detailed descriptions of the settings, see [“Client policy attributes reference”](#) on page 407 or [“Certificate definition policy attributes reference”](#) on page 432.)

To return to the basic display, click **Hide Descriptions**.

You have now displayed a user policy.

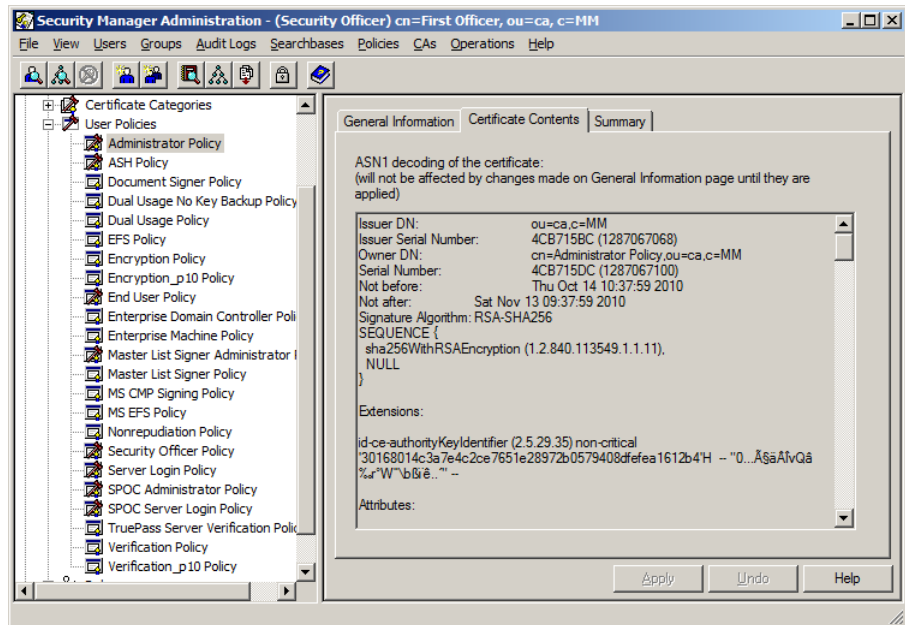
- 6** (Optional.) If you want to save the summary for future reference, you can select and copy the summary into a text editor or spreadsheet and then save it to a file. The information displayed is delimited by tabs for easy viewing in a spreadsheet.

# Viewing policy certificates

You can view the ASN.1 decoding of user policy certificates.

## To view a policy certificate

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration”](#) on page 46).
- 2 In the tree view, expand **Security Policy > User Policies**.



- 3 Select the user policy that you want to view.
- 4 Click the **Certificate Contents** tab to view the ASN.1 decoding of the policy certificate.

# Creating user policies

Security Manager includes several default user policies that you can use in your PKI system (see [“Predefined user policies” on page 393](#)). If you require more user policies than the default user policies included with Security Manager, you can create user policies.

To create user policies, you must have sufficient permissions. By default, only Security Officers can create user policies. To create a user policy, you can create a new user policy, or you can copy an existing user policy and then change the user policy's settings as required. Create a new user policy when you want a new unique user policy. Copy an existing user policy when you want a new user policy that is similar to an existing user policy.

To assign user policies to roles, see [“Modifying roles” on page 363](#). To map user policies to certificate definitions, see [“Mapping policy certificates to certificate definitions” on page 404](#).

All available user policy attributes are defined in a certificate specifications file. You can create additional user policy settings by editing this file. It is strongly recommended that you do not alter the properties of any of the default attributes. For more information, see [“Customizing policy certificates” on page 571](#).

This section contains the following procedures:

- [“To create a new user policy” on page 398](#)
- [“To create a user policy by copying an existing user policy” on page 400](#)

## To create a new user policy

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Select **Policies > User Policies > New**.  
The **New User Policy** dialog box appears.

**New User Policy**

Label:

Common name:

Add to: CA Domain Searchbase

Category: Policy Certificates

Type: Client Settings (Client Setting Certificates)  
 Cert. Defn. Settings (Policy Settings for Certificate Definitions)

Policy Attributes

Password expires in (weeks):

Password history:

Password length (characters):

Password needs non-alpha char.: ☐

OK Cancel Help

- 3** In the **Label** field, enter a name that describes the new user policy.
- 4** In the **Common name** field, enter the common name for the new policy certificate's directory entry (the `cn=` value of the entry). Do not include `cn=` when entering the common name.
- 5** In the **Add to** drop-down list, select the searchbase where you want to store the user policy. (For more information about searchbases, see ["Administering searchbases" on page 341.](#))  
 Click **Show DN** to view the distinguished name (DN) of the user policy's directory entry. If you use Active Directory, the entry must already exist in the directory.
- 6** In the **Type** box, select **Client Settings** to create a client user policy, or select **Cert. Defn. Settings** to create a certificate definition user policy.  
 The certificate type you choose determines which policy attributes you can set. For more information about user policy types, see ["User policy overview" on page 392.](#)
- 7** Under **Policy Attributes**, configure the user policy attributes as required.

For more information about each attribute, see [“Client policy attributes reference” on page 407](#) or [“Certificate definition policy attributes reference” on page 432](#).

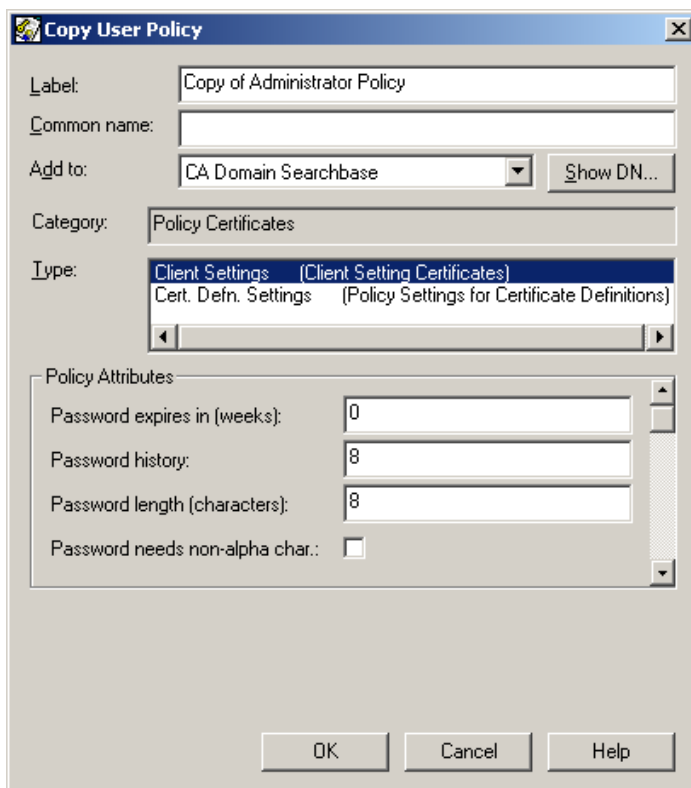
- 8** Click **OK**.
- 9** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

### To create a user policy by copying an existing user policy

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** In the tree view, expand **Security Policy > User Policies**.
- 3** Select the user policy that you want to copy and then select **Policies > User Policies > Selected User Policy > Copy**.

The **Copy User Policy** dialog box appears.



The screenshot shows the "Copy User Policy" dialog box. It contains the following fields and options:

- Label:** Copy of Administrator Policy
- Common name:** (empty field)
- Add to:** CA Domain Searchbase (dropdown menu) with a "Show DN..." button.
- Category:** Policy Certificates
- Type:** Client Settings (Client Setting Certificates) (dropdown menu). Other options visible are "Cert. Defn. Settings (Policy Settings for Certificate Definitions)".
- Policy Attributes:**
  - Password expires in (weeks): 0
  - Password history: 8
  - Password length (characters): 8
  - Password needs non-alpha char.: ☐
- Buttons:** OK, Cancel, Help

- 4** In the **Label** field, enter a name that describes the new user policy.



- 5 In the **Common name** field, enter the common name for the new policy certificate's directory entry (the `cn=` value of the entry). Do not include `cn=` when entering the common name.
- 6 In the **Add to** drop-down list, select the searchbase where you want to store the user policy. (For more information about searchbases, see ["Administering searchbases" on page 341.](#))  
Click **Show DN** to view the distinguished name (DN) of the user policy's directory entry. If you use Active Directory, the entry must already exist in the directory.
- 7 If you want to change the user policy type, then select a new user policy type in the **Type** box. Select **Client Settings** to create a client user policy, or select **Cert. Defn. Settings** to create a certificate definition user policy.  
The certificate type you choose determines which policy attributes you can set. For more information about user policy types, see ["User policy overview" on page 392.](#)
- 8 Under **Policy Attributes**, configure the user policy attributes as required.  
For more information about each attribute, see ["Client policy attributes reference" on page 407](#) or ["Certificate definition policy attributes reference" on page 432.](#)
- 9 Click **OK**.
- 10 If prompted to authorize the operation, authorize the operation. See ["Authorizing sensitive operations" on page 52.](#)  
If the operation was successful, a success message appears.

# Modifying user policies

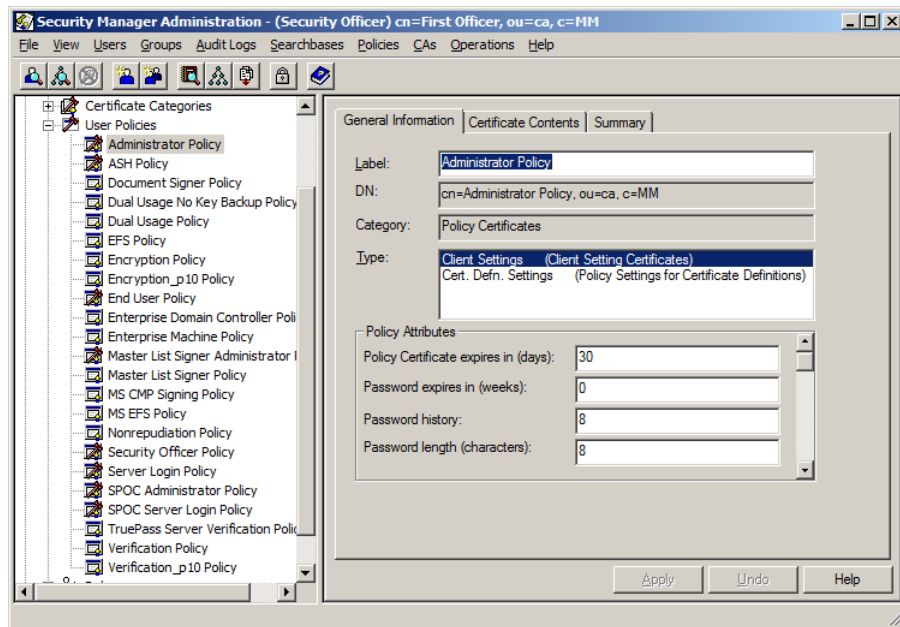
You can change the values of any user policy settings through Security Manager Administration. The settings for each user policy are contained within a policy certificate. Like other certificates, all policy certificates are stored in the Security Manager directory. When you make changes to a user policy, affected users receive an updated user policy certificate the next time they log in to a Security Manager client application or during their next key management operation.

All available user policy attributes are defined in a certificate specifications file. You can create additional user policy attributes by editing this file. It is strongly recommended that you do not alter the properties of any of the default attributes. For more information, see [“Customizing policy certificates” on page 571](#).

To assign user policies to roles, see [“Modifying roles” on page 363](#). To map user policies to certificate definitions, see [“Mapping policy certificates to certificate definitions” on page 404](#).

## To modify a user policy

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 In the tree view, expand **Security Policy > User Policies**.



- 3 Select the user policy that you want to modify.
- 4 Click the **General Information** tab.

- 5 If you want to change the name of the user policy, enter a new name in the **Label** field.

---

**Attention:** Changing the user policy type impacts all Security Manager clients currently using the user policy. Security Manager Administration warns you if you attempt to change the policy type. Ensure that you really do intend to change the user policy type before clicking **Yes** to change the user policy type.

---

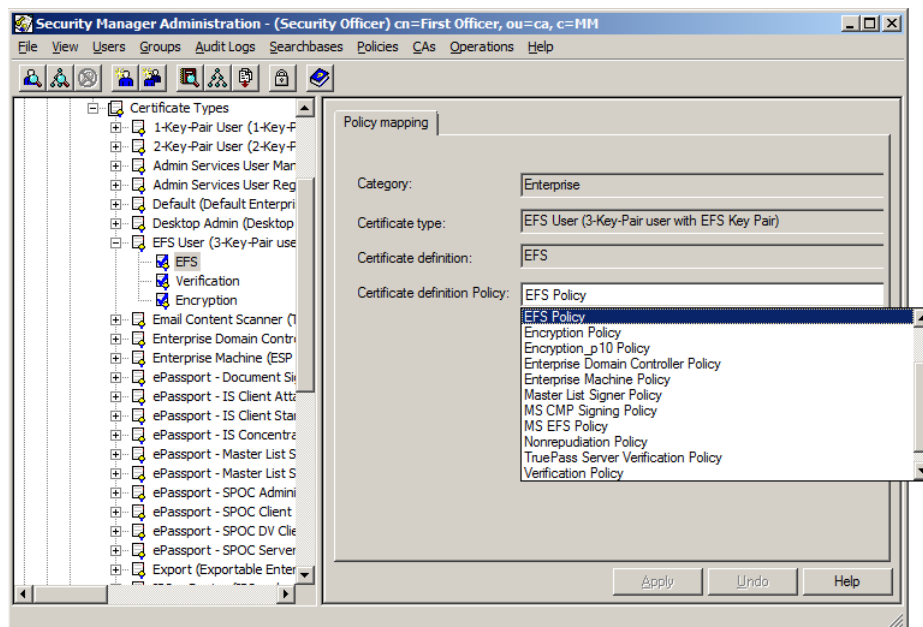
- 6 If you want to change the user policy type, select a new user policy type in the **Type** box. For more information about user policy types, see [“User policy overview” on page 392](#).
- 7 Under **Policy Attributes** to configure the policy attributes as required.  
For more information about each policy attribute, see [“Client policy attributes reference” on page 407](#) or [“Certificate definition policy attributes reference” on page 432](#).
- 8 Click **Apply** when you have finished making changes.
- 9 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
If the operation was successful, a success message appears.
- 10 (Optional.) If you want to view the ASN.1 decoding of the policy certificate and see the results of your changes, see [“Viewing policy certificates” on page 397](#).

# Mapping policy certificates to certificate definitions

After defining certificate definitions attributes for a user policy, you can map the policy certificate to a certificate definition. This process associates the certificate definition settings with the certificate definition, so that every user with that certificate is governed by the associated policies.

## To map a policy certificate to a certificate definition

- 1 Log in to Security Manager Administration as a Security Officer (see [“Logging in to Security Manager Administration”](#) on page 46).
- 2 Expand **Security Policy > Certificate Categories**.
- 3 Expand the certificate category (for example, **Enterprise**).
- 4 Expand **Certificate Types**.
- 5 Expand the required certificate type (for example, **EFS User**), and then select the certificate definition (for example, **EFS**).



- 6 In the **Certificate definition policy** drop-down list, select the appropriate certificate definition policy and then click **Apply**.
- 7 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations”](#) on page 52.

If the operation was successful, a success message appears.

# Deleting user policies

Delete a user policy if it is no longer required. Before doing so, ensure that it is no longer mapped to a role or certificate definition.

## To delete a user policy

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** In the tree view, expand **Security Policy > User Policies**.
- 3** Select the user policy that you want to delete.
- 4** Select **Policies > User Policies > Selected User Policy > Delete**.
- 5** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

If the operation was successful, a success message appears.

# Importing and exporting user policies

You can choose to export user policies from Security Manager so that you can import them to another Certification Authority (CA).

## To export a user policy

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 In the tree view, expand **Security Policy > User Policies**.
- 3 Select the user policy you want to export
- 4 Select **Policies > User Policies > Selected User Policy > Export**.  
The **Save As** dialog box appears.
- 5 Save the policy to a file with a `.userpolicy` extension.
- 6 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
If the operation was successful, a success message appears.
- 7 Copy the file onto the computer hosting the other CA.

## To import a user policy

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Select **Policies > User Policies > Import**.  
The **Open** dialog box appears.
- 3 Locate the file and then click **Open**.  
The **Import User Policy** dialog box appears.
- 4 Make changes to the user policy as required. For details, see [“Modifying user policies” on page 402](#).  
You must enter a common name in the **Common Name** field, and select the appropriate searchbase in the Add to drop-down list. If you use Active Directory, the directory entry must already exist.
- 5 Click **OK**.
- 6 If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
If the operation was successful, a success message appears. The imported policy now appears in the list of user policies. In the tree view, expand **Security Policy > User Policies** to see the imported policy in the list of user policies.

# Client policy attributes reference

Client policies define the certificates associated with roles, so that the policy attributes you define applies to every user with the associated role. For example, when you define attributes for the End User role, every with the End User role is governed by those policy attributes.

This section describes all the policy attributes that you can configure in the default client policies.

## Policy Certificate expires in (days)

The **Policy Certificate expires in (days)** setting allows you to specify how long a user policy (which is defined in a policy certificate) is valid.

Every time the user logs in to Security Manager, the user's policy certificate refreshes. If any changes are made to the policy certificate, these are picked up by the user when the user logs in.

Users who are working offline can continue to work offline for as long as their policy certificate is valid. The longer a policy certificate is valid, the longer users associated with that policy certificate can work offline. At the same time, the longer a policy certificate is valid, the longer you are allowing users to work offline with potentially out-of-date user policies. You must strike a balance between allowing users to work offline and maintaining tight control of your user policies. In extreme cases, you could limit offline use to as little as one day by setting this to 1. Doing so forces users to log in online at least once in every 24 hours.

You can set the **Policy Certificate expires in (days)** setting from 1 to 3650 days. By default, all policy certificates expire after 30 days.

## Password expires in (weeks)

The **Password expires in (weeks)** setting designates the length of time a user's Entrust password is valid. When the set amount of time transpires, the password expires and the user is prompted to create a new password.

You can set password lifetimes from 0 to 52 weeks, or you can set the password lifetime to zero (0), which means the password never expires. By default, **Password expires in (weeks)** is set to 0 (the password never expires).

## Password history

The **Password History** setting designates the number of unique passwords a user must create before the user can reuse an old password. If a user is creating a new password, and it matches any of the passwords stored in the password history, the user is prompted to create a different password.

---

**Note:** When you recover a user's keys, the user's Entrust desktop application deletes the user's profile if it still exists. As a result, the user's password history, which is stored in the profile, is no longer available. This means that a recovered user can choose the same password for the new profile as the previous profile.

---

You can set the password history from 0 to 8 passwords. By default, **Password History** is set to 8 passwords.

## Password length (characters)

The **Password length (characters)** setting designates the minimum number of characters that must appear in a user's password.

You can set the minimum password length from 8 to 20 characters. By default, **Password length (characters)** is set to 8 characters.

---

**Attention:** A minimum password length of eight characters is recommended. Shorter passwords may significantly weaken end user security in certain environments. For further information on suitable environments for a lesser value, contact Entrust Support.

---

## Password needs non-alpha char.

The **Password needs non-alpha char.** setting is a true-or-false setting that designates whether users must include a non-alphanumeric character in their passwords.

Non-alphanumeric characters include, but are not limited to, the following: ! @ # \$ % ^ & \* .

By default, users do not require a non-alphanumeric character in their passwords (the **Password needs non-alpha char.** setting is disabled).

## Password needs uppercase letter

The **Password needs uppercase letter** setting is a true-or-false setting that designates whether users must include an uppercase letter in their passwords.

By default, users require an uppercase letter in their passwords (the **Password needs uppercase letter** setting is enabled).

## Password needs lowercase letter

The **Password needs lowercase letter** setting is a true-or-false setting that designates whether users must include a lowercase letter in their passwords.



By default, users require a lowercase letter in their passwords (the **Password needs lowercase letter** setting is enabled).

## Password needs number

The **Password needs number** setting is a true-or-false setting that designates whether users must include a number in their passwords.

By default, the **Password needs number** setting is disabled.

## Disable single login

The **Disable single login** setting is a true-or-false setting that designates whether end users can use the single login feature.

When the **Disable single login** setting is enabled, users are required to log in to every Entrust desktop application they launch. When the **Disable single login** setting is disabled, users can log in once to an Entrust desktop application, and have Security Manager automatically secure each subsequent Entrust desktop application they launch, without requiring the users to enter their Entrust password again.

By default, only End Users can use the single login feature (the **Disable single login** setting is disabled). Entrust PKI administrators are not allowed to use single login due to security reasons (that is, the **Disable single login** setting is enabled in the Security Officer Policy and the Administrator Policy). Single login is also disabled for the Server Login Policy.

## Login timeout (minutes)

The **Login timeout (minutes)** setting designates how many minutes of inactivity must pass before the user is automatically logged out of an Entrust desktop application. You can set the **Login timeout (minutes)** setting from 0 to 300 minutes.

In some cases, you may not want Entrust desktop applications to time out. For example, an Entrust desktop application that prints cheques automatically stops running if the application times out. If you want Entrust desktop applications to not time out, set the **Login timeout (minutes)** setting to 0. Note, however, that setting **Login timeout (minutes)** to 0 does not apply to Security Manager Administration. If **Login timeout (minutes)** is set to 0, Security Manager Administration times out according to the defaults.

By default, **Login timeout (minutes)** is set to 10 minutes in the Security Officer Policy and the Administrator Policy, and in the End User Policy the default is set to 15 minutes.

---

**Note:** Users of Entrust desktop applications can choose a timeout that is less than or equal to the **Login timeout (minutes)** setting specified in the End User Policy.

---

For more information on this setting, see the information on password management in the Entrust desktop application documentation.

## Symmetric encryption algorithms

The **Symmetric encryption algorithms** setting designates which algorithms users can employ for encrypting data. When you configure this setting, it is important to keep in mind which algorithms the client application supports.

You must include in the **Symmetric encryption algorithms** setting at least one algorithm that the client application supports. For example, if you specify IDEA-128 as the only allowed symmetric encryption algorithm, and the client application does not support IDEA-128, users are unable to encrypt.

Enter any of the following algorithms exactly as shown into the **Symmetric encryption algorithms** text field:

- ALL
- CAST-128
- CAST-80
- CAST-64
- CAST-40
- IDEA-128
- AES-256
- TRIPLE-DES
- DES
- RC2-128Compatible
- RC2-40Compatible.

(Use spaces instead of commas when entering values into the **Symmetric encryption algorithms** text field.)

The value ALL allows users to encrypt using any of the algorithms from this list. If the value ALL is specified, the default encryption algorithm is CAST-128.

If you enter more than one value into the **Symmetric encryption algorithms** field, the first value entered is taken as the default. For example, if you enter the values CAST-128 TRIPLE-DES IDEA-128 into the **Symmetric encryption algorithms** field, CAST-128 is set as the default encryption algorithm, and CAST-128, TRIPLE-DES, and IDEA-128 are the algorithms users are permitted to use for encrypting data. A user

can overwrite this default as long as the algorithm chosen by the user matches one of the permitted algorithms listed in the **Symmetric encryption algorithms** field.

## Key type for signatures

The **Key type for signatures** setting specifies the type of user signing, nonrepudiation, and dual usage keys to be generated by V1 client applications. For V2 client applications, this setting is superseded by the **Algorithm for key pair** value in the certificate definition policy (see [“Algorithm for key pair” on page 440](#)).

When you configure this setting, it is important to consider which key types that the client application supports. For example, if you specify DSA-1024 as the key type, and the client application does not support DSA-1024, key management will fail.

Using uppercase letters only, type one of the following key types in the **Key type for signatures** field:

- RSA-1024
- RSA-2048
- RSA-3072
- RSA-4096
- RSA-6144
- DSA-1024
- ECDSA-192

This key type is supported for backwards compatibility. It is the same as EC-P-192.

- EC-<curve>

Where <curve> is a named elliptic curve. For a list of supported elliptic curves, see the *Security Manager Operations Guide*.

When you are creating a user policy for 1-key-pair users (users who use this key type for both signing and encryption), do not specify DSA-1024. Encryption is not supported with DSA. For more information, see [“Number of key pairs” on page 423](#).

---

**Note:** When you select **Enable CAPI Synchronization**, you can only choose an RSA or DSA value as the policy setting for **Key type for signatures**. ECDSA-192 and EC algorithms are not supported for CAPI profile export. If you choose ECDSA-192 or an EC algorithm, users you create with this user policy are unable to use CAPI export. The **Enable CAPI Synchronization** policy setting is ignored, and no explanatory message is issued. See [“Enable CAPI Synchronization” on page 422](#).

---

## Key type for encryption

The **Key type for encryption** setting specifies the type of user encryption keys to be generated for V1 client applications. For V2 client applications, this setting is superseded by the **Algorithm for key pair** value in the certificate definition policy (see [“Algorithm for key pair” on page 440](#)).

When you configure this setting, it is important to consider which key types the client application supports. For example, if you specify RSA-3072 as the key type, and the client application does not support RSA-3072, key management will fail.

Using uppercase letters only, type one of the following key types in the **Key type for encryption** field:

- RSA-1024
- RSA-2048
- RSA-3072
- RSA-4096
- RSA-6144
- ECDSA-192

This key type is supported for backwards compatibility. It is the same as EC-P-192.

- EC-<curve>

Where <curve> is a named elliptic curve. For a list of supported elliptic curves, see the *Security Manager Operations Guide*.

---

**Note:** When you select **Enable CAPI Synchronization**, you can only choose an RSA value as the policy setting for **Key type for encryption**. ECDSA-192 and EC algorithms are not supported for CAPI profile export. If you choose ECDSA-192 or an EC algorithm, users you create with this user policy are unable to use CAPI export. The **Enable CAPI Synchronization** policy setting is ignored, and no explanatory message is issued. See [“Enable CAPI Synchronization” on page 422](#).

---

## Message in Entrust-Ready clients

The **Message in Entrust-Ready clients** setting allows you to present a message to users when they log in to Security Manager using the Entrust Login Interface (part of Entrust Desktop Solutions). This setting is ideal for conveying important Security Manager system administration information to end users, who see the message as soon as they log in.

By default, the **Message in Entrust-Ready clients** setting is blank.

## Permit roaming

The **Permit roaming** setting designates whether users can download their profile from the Roaming Server. When **Permit roaming** is enabled, users can access Entrust desktop applications by downloading their user profile from the Roaming Server, regardless of where they are located or what machine they are using. When **Permit roaming** is disabled, users can only access Entrust desktop applications with a user profile that is stored locally on disk or on a token. If the **Permit roaming** setting is disabled, you must enable the **Permit desktop** setting.

By default, the **Permit roaming** setting is enabled. If you disable it, you must enable the **Permit desktop** setting. (You must make sure that one of the **Permit roaming** setting or the **Permit desktop** setting is enabled.)

For more information, see the *Entrust Authority Roaming Server Administration Guide*.

## Permit desktop

The **Permit desktop** setting designates whether users have a user profile that is stored locally on disk or on a token.

You may wish to disable the **Permit desktop** setting for roaming users, because it reduces the risk of roaming users losing their locally stored user profile when they are working outside the office.

By default, the **Permit desktop** setting is enabled. If you disable it, you must enable the **Permit roaming** setting. (You must make sure either the **Permit roaming** setting or the **Permit desktop** setting is enabled.)

## Enforce token usage

The **Enforce token usage** setting designates whether users must use a hardware token to access Entrust desktop applications.

Enforcing token usage increases the level of access control to Entrust desktop applications. When the **Enforce token usage** setting is enabled, users must have possession of a hardware token, as well as knowledge of their Entrust password, in order to access Entrust desktop applications. For example, you might want to enforce token usage for all Entrust PKI administrators who have access to the Security Manager Administration application.

If the **Enforce token usage** setting is enabled, you must also enable the **Permit desktop** setting. By default, the **Enforce token usage** setting is disabled.

If **Number of key pairs** is set to 1, you must disable **Enforce token usage**. The use of hardware tokens is not supported for 1-key-pair users. If you enable this setting for a 1-key-pair user, the user is unable to create an Entrust profile. See [“Number of key pairs” on page 423](#).

If **Allow PKCS#12 Export** is enabled, you must disable **Enforce token usage**. Users who use a hardware token cannot export their profiles in PKCS #12 format. See [“Allow PKCS#12 Export” on page 421](#).

## Allow personal addr. book use

The **Allow personal addr. book use** setting designates whether users can maintain personal trust relationships with individual users inside their CA domain, including users in cross-certified CAs, and outside their domain. These relationships are maintained using a personal address book.

By default, the **Allow personal addr. book use** setting is enabled.

## Allow CA personal addr. book use

The **Allow CA personal addr. book use** setting applies to the IPSec Toolkit. This setting designates whether users can maintain personal trust relationships with other, uncross-certified CA domains. Allowing users to maintain a personal trust relationship with an uncross-certified CA domain means you allow your users to trust all the users in a CA domain with which your CA is not cross-certified. These relationships are maintained using a Certification Authority personal address book.

By default, the **Allow CA personal addr. book use** setting is disabled.

## Permit Server Login usage

Enable this setting to permit users to create the encrypted password file for use with Server Login. Users who currently use this file can continue to use Server Login as long as they do not change their computer or password.

By default, this setting is disabled.

## Enforce identity usage

When enabled, this setting lets users create their Entrust profile using an identity device.

By default, this setting is disabled.

## DN encoding formats

The **DN encoding formats** setting designates what format client applications use to encode certain attributes of the DN. Only those attributes that are defined with the `directoryString` syntax (described in the X.520 standard) are encoded using the specified format. DN attributes are encoded, for example, during S/MIME or PKIX-CMP operations.

When DN encoding takes place, if the user policy specifies `printable`, and the attribute values that require encoding fall within the `PrintableString` range of characters (that is, a-z A-Z 0-9 ' ( ) + , - . / : = ? space) then `PrintableString` is used.

If a character falls outside this range, and `teletex` is specified, the client application attempts to encode using `TeletexString` (a character set that fits within the Latin-1 range). If `teletex` is not specified, or a character falls outside its range, the Entrust desktop application attempts to encode using UTF8, if it is specified (UTF8 is a special encoding of Unicode, and is the preferred method for encoding format). If UTF8 is not specified, the client application attempts to encode using `BMPString`, if it is specified (the `BMPString` format is another form of Unicode). If a character falls outside the range of `PrintableString` and `TeletexString`, and neither UTF8 nor `BMPString` is specified, the Entrust desktop application reports an error.

The **DN encoding formats** setting allows you to disable formats that are now deprecated, although you may need to maintain deprecated formats to remain compatible with legacy systems. The formats that are currently deprecated are `teletex` and `bmp`.

By default, the **DN encoding formats** setting supports `printable`, `teletex`, and `utf8`.

## Perform dir. consistency check

Enable this setting for a directory consistency check. By default, the **Perform dir. consistency check** setting is enabled.

The **Perform dir. consistency check** setting designates whether the DN in a certificate is checked against the DN of the directory entry that contains that certificate. Each certificate has a field (the subject name field) that uses a DN to name the subject of the certificate. DNs are also used to designate the location of each entry in a directory. Often, these two DNs have the same value. However, there are several situations in which these two values could differ.

For example, a local CA service provider may have a public directory that lists all the organizations in your area, and you might find your certificate located in this directory under `cn=your name, ou=your company, ou=software companies, dc=your town's public directory`. This DN indicates the structure of the public directory, and indicates the location of your entry in that directory. However, your company's CA generated your certificate, and the DN in your certificate's subject name field is different from the DN of your entry in the public directory. For example, the subject name DN might be `cn=your name, ou=marketing, dc=Company One, dc=com`. In a case like this, or in any other case where the DN of the directory entry is different from the DN in the certificate subject name field, you must disable the **Perform dir. consistency check** setting.

When the **Perform dir. consistency check** setting is enabled, the DN of the directory entry is compared against the DN of the certificate subject name. If, for a given user, the values of these two DNs are different, then you cannot encrypt for that user.

## Management Client

Allows you to set whether a Card Management System or an Entrust client application such as Security Provider for Windows or Security Toolkit for Java Platform writes certificate management tasks to a card.

Use this policy to store managed digital IDs on a write-protected card. This option indicates to the Entrust client application whether it should attempt to write credential information to the card itself, or whether it should communicate with the Card Management System to do so.

Possible values are `Entrust` or `CardMS`.

By default, this setting is set to `Entrust`.

## Force Original CD Compliance

Forces a client such as Security Provider for Windows to honor the original certificate definition policies on all certificates in a digital ID's history. If false, a client can override policies on historical certificates. This is useful in the following scenarios:

- You have smart card users, and a smart card user forgets the card at home. The corporate procedure recovers the user to a temporary EPF file for that day. When the user brings the smart card, the user recovers again onto the card.
- You use smart cards from another company, but then decide to buy the Entrust iKey token instead.

All users need to change from the one card to the other. During recovery, the new keys and the key history are put on the Entrust iKey token. The old smart card is no longer required.

By default, this setting is set to false.

## Enforce S/MIME

The **Enforce S/MIME** setting designates whether you are required to communicate using S/MIME (Secure Multipurpose Internet Mail Extensions) when using Entrust Entelligence E-Mail Plug-in (formerly Entrust/Express).

By default, the **Enforce S/MIME** setting is disabled.

## Allow S/MIME Secure Receipts

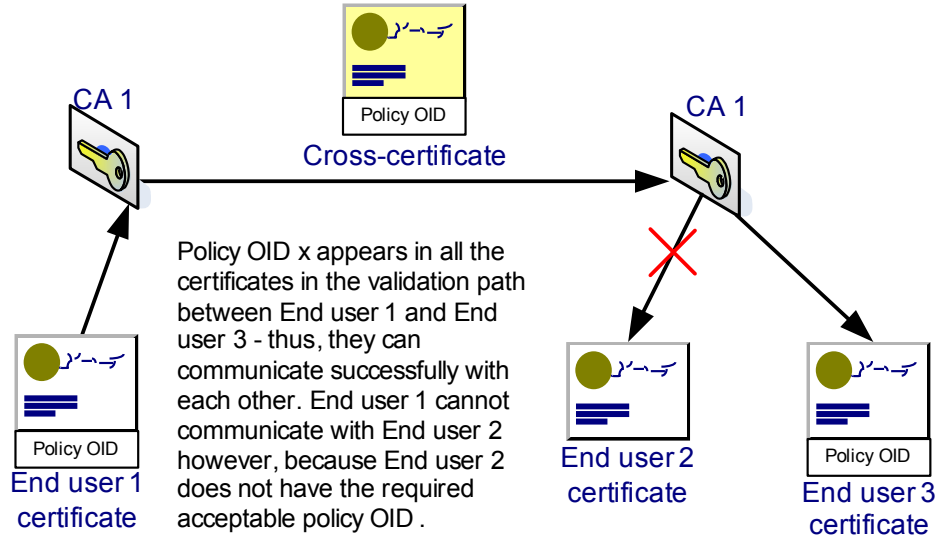
The **Allow S/MIME Secure Receipts** setting allows you to request the return of a secure receipt when sending S/MIME formatted email messages using Entrust Entelligence E-Mail Plug-in (formerly Entrust/Express).

By default, the **Allow S/MIME Secure Receipts** setting is disabled.



## Acceptable policy OIDs

The **Acceptable policy OIDs** setting allows you to enforce policies for users in your CA domain. You can use policy OIDs for just about anything, such as dictating how users communicate with each across different CA domains.



For example, a CA that you are cross-certified with might have user policies with varying grades of security for its different end-user roles. To add a user to a role with a user policy of high security, the user must go through a security check. Users who do not go through a security check are assigned to a role with a lower level of security in the user policy. Therefore, the policy of this CA dictates that users must undergo a security check before they can be highly trusted.

This policy is represented numerically as an OID (that is, an object identifier). For example, suppose the OID for this policy is 2.6.7.1.47.98.2. An Entrust PKI administrator can include policy OID 2.6.7.1.47.98.2 in the encryption certificate, the verification certificate, or both certificates of highly trusted users. The Entrust PKI administrator can then designate policy OID 2.6.7.1.47.98.2 as a policy that must be checked for. This is done by including the OID in the **Acceptable policy OIDs** setting of the user policy for highly trusted users. The user policies of lower security, on the other hand, would not have this OID included in the **Acceptable policy OIDs** setting.

You might decide that users belonging to a highly trusted role in your CA domain should only communicate with highly trusted users in the other CA domain. To enforce this measure for your highly trusted users, you select the **Require policy** check box in the **End User Policy**, enter policy OID 2.6.7.1.47.98.2 in the **Acceptable policy OIDs** field, and use the certificate specification file to add policy OID 2.6.7.1.47.98.2 to your CA's cross-certificate. (The cross-certificate makes up a part

of the validation path between users in different CA domains, so the cross-certificate must also contain the required policy OID.) Now users in your CA domain can only communicate successfully with users in the other CA domain that also have policy OID 2.6.7.1.47.98.2 in their policy certificate.

By default, no policy OIDs are included in the **Acceptable policy OIDs** setting. To enforce checking of policy OIDs, you must enable the **Require policy** setting (see [“Require policy” on page 418](#)). Policy OIDs can have from 0 to 32 arcs. The arcs are the sets of numbers between the decimal points. For example, our example policy OID, 2.6.7.1.47.98.2, has seven arcs. For more information on OIDs, see the *Security Manager Directory Configuration Guide*.

For information on adding policy OIDs to cross-certificates, see [“Customizing cross-certificates” on page 580](#).

## Do not process policy mappings

The **Do not process policy mappings** setting disables policy mapping.

Policy mapping can take place between organizations that have the same policy, but use different policy OIDs to represent this policy. For example, Company One may have a policy that requires highly trusted users to undergo a security check. CA domain A represents this policy using policy OID 2.6.7.1.47.98.2. CA domain B has the same policy, but represents it using policy OID 2.6.7.1.47.98.4. It is possible for CA domain A to map OID 2.6.7.1.47.98.2 to OID 2.6.7.1.47.98.4 so users in CA domain A can communicate with users in CA domain B. However, CA domain B may decide to disable policy mapping to ensure stricter security. You can choose to enable or disable Policy mapping using the **Do not process policy mappings** setting.

By default, the **Do not process policy mappings** setting is disabled.

## Require policy

The **Require policy** setting works in conjunction with the **Acceptable policy OIDs** setting. If you want Entrust desktop applications to check for and validate the list of acceptable policy OIDs, you must enable the **Require policy** setting.

By default, the **Require policy** setting is disabled.

## PKIX compliance

The **PKIX compliance** setting allows you to control the set of certificates relied upon by your users with respect to PKIX compliance. If the **PKIX compliance** setting is enabled, Security Manager performs the following checks on each certificate that your users rely on:

- The Security Manager key identifier extension is present in all certificates, is non-critical, and contains a key identifier field.

- If the basic constraints extension is present and indicates that the subject is a CA, the subject key identifier extension is present and non-critical.
- The `basicConstraints` extension is present in all certificates except possibly the last certificate in the chain. It is also present in all certificates containing a key usage extension with the `keyCertSign` bit set. The `basicConstraints` extension, if present, is critical.
- The basic constraints extension, if present, has `cA` set to `true`.
- If the key usage extension is present and the `keyCertSign` bit is set, the certificate contains the basic constraints extension with `cA=true`.
- If the private key usage period extension is present, it is non-critical and contains at least one field (either `notBefore` or `notAfter`).
- The policy mappings extension, if present, is non-critical.
- The policy mappings extension appears only in certificates with a basic constraints extension having `cA=true`.
- The name constraints extension appears only in CA certificates, and is set to `critical`.
- The policy constraints extension appears only in CA certificates, and, if present, contains at least one of `requireExplicitPolicy` or `inhibitPolicyMapping`.

Should one or more of these checks fail, users cannot rely on the certificate because it is non-PKIX compliant.

By default, the **PKIX compliance** setting is disabled.

## FPKI compliance

The **FPKI compliance** setting allows you to control the set of certificates that your users rely on with respect to FPKI compliance. If the **FPKI compliance** setting is enabled, Security Manager performs the following checks on each certificate that your users rely on:

- The authority key identifier extension is present in all certificates except the self-signed certificate, and is non-critical.
- The authority key identifier extension, if present, contains a key identifier field, but no `authorityCertIssuer` or `authorityCertSerialNumber`.
- The subject key identifier extension is present in all certificates and is non-critical.
- The key usage extension is present in all certificates except the self-signed certificate.
- The key usage extension, if present, is critical and is set to one of the allowed combinations.

- If the private key usage period extension is present, it is non-critical and contains at least one field (either `notBefore` or `notAfter`).
- The certificate policies extension is present in all certificates except possibly the self-signed certificate.
- The policy mappings extension, if present, is non-critical.
- The policy mappings extension appears only in certificates with a basic constraints extension having `cA=true`.
- The basic constraints extension is present in all certificates. In the self-signed certificate it is non-critical. In all other certificates, it is critical. In all but possibly the last certificate in the chain, `cA` must be `true`.
- The name constraints extension appears only in CA certificates, does not appear in the self-signed certificate, and, if present, is set to critical.
- The policy constraints extension appears only in CA certificates and must be critical. It does not appear in the self-signed certificate.
- The CRL distribution points extension does not appear in the self-signed certificate.

Should one or more of these checks fail, users cannot rely on the certificate because it is non-FPKI compliant.

By default, the **FPKI compliance** setting is disabled.

## Do not process anyPolicy

Use the **Do not process anyPolicy** setting to control the handling of the `any-policy` certificate policy identifier. As long as `any-policy` is not inhibited, the identifier is considered a match for any policy. When the `any-policy` policy identifier is inhibited, it is not considered a match for other policies when processing the certificate policies extension, unless the certificate in which it appears is self-issued. Use this setting to inhibit `any-policy` at the beginning of certificate validation.

By default, this setting is false (disabled).

## HTTP Proxy for CRL Requests

The HTTP Proxy is an `http` address that sends HTTP requests for CRLs. It defines the HTTP proxy used to retrieve CRLs through HTTP.

For example, if this is set to `http://crlserver.company.com/`, and a certificate has a CRL Distribution point showing the CRL at

`http://crl.somewhereelse.com/crlfile.crl`, then the Entrust client software sends a request to `http://crlserver.company.com/` for

`http://crl.somewhereelse.com/crlfile.crl`. If this setting is not present, the client software sends the request for `crlfile.crl` directly to

`http://crl.somewhereelse.com/`.

Note that this setting is only available after the policy certificate is verified, meaning that it is possible that the proxy setting is not available for the first CRL requests. By default, this setting is disabled (blank).

## Allow PKCS#12 Export

The **Allow PKCS#12 Export** setting allows you to specify whether users can export their profile in PKCS #12 format.

Note that before users can export their profile, one or both of their user certificates must also contain an extension allowing profile export. See [“Allowing profile export” on page 244](#).

If **Allow PKCS#12 Export** is enabled, you must disable **Enforce token usage**. Users who use a hardware token cannot export their profiles in PKCS #12 format. See [“Enforce token usage” on page 413](#).

## All Exportable

The **All Exportable** setting allows all non-exportable key pairs to be retroactively marked as exportable. If this setting is selected, it overrides the **Allow PKCS#12 Export** setting.

Note that before users can export their profile, one or both of their user certificates must also contain an extension allowing profile export. See [“Allowing profile export” on page 244](#).

If **All Exportable** is enabled, you must disable **Enforce token usage**. Users who use a hardware token cannot export their profiles in PKCS #12 format. See [“Enforce token usage” on page 413](#).

By default, this setting is disabled (blank).

## Minimum PKCS#12 Hash Count

The **Minimum PKCS#12 Hash Count** setting allows you to specify the minimum number of times that the user-supplied password for the PKCS #12 file is hashed during export. A value of 1 for this setting is very weak, 2000 is acceptable, and 10000 is highly secure.

Note that the default minimum hash count setting (2000) may prevent older browsers from reading the PKCS #12 file. If you want to allow users to specify a lower hash count so that older browsers can read the exported files, consider setting the minimum hash count to 1. More recent browsers support either a hash count of 2000 or variable hash counts. By default, this setting is 2000.

## Enable CAPI Synchronization

The **Enable CAPI Synchronization** setting allows a user to export a profile to Microsoft CAPI (Cryptographic Application Programming Interface). This feature allows a user to synchronize the profile with CAPI so that the user can import certificates and private keys into the Microsoft CAPI store. The user's certificates and private keys are imported to the CAPI store for use with CAPI-enabled applications (such as Microsoft Outlook Express and Microsoft Internet Explorer).

If this setting is enabled and **Unprotected CAPI key storage?** is disabled, the user is prompted to select a level of protection for the exported keys.

If this setting is enabled, you can also enable the **Private key export from CAPI?** (see [“Private key export from CAPI?” on page 423](#)).

Note that before a user can export a profile, one or both of the user's certificates must also contain an extension allowing profile export. See [“Allowing profile export” on page 244](#).

When you enable **Enable CAPI Synchronization**, you can only set the for **Key type for signatures** and **Key type for encryption** settings to an RSA or DSA value. ECDSA-192 and EC algorithms are not supported for CAPI profile export. If you choose ECDSA-192 or an EC algorithm, users you create with this user policy cannot create an Entrust profile. See [“Key type for signatures” on page 411](#) and [“Key type for encryption” on page 412](#).

CAPI profile export is supported for roaming users, but is not recommended in all situations. Roaming typically mandates a zero footprint on the client machine, while CAPI is based on local storage of certificates and keys. Whenever an Entrust PKI user logs in using a roaming profile, CAPI export executes. The certificates and keys imported into the CAPI store remain once the user logs out.

When deciding whether to enable CAPI export for a roaming user, consider the following:

- If the roaming user consistently logs in to a personal Windows account on one or more machines, and the operating system is Windows 2000, then it is acceptable to allow CAPI export for the user.
- If roaming users are sharing Windows accounts (in a kiosk situation or on home computers) or are using an operating system other than Windows 2000, Entrust does not recommend allowing CAPI export. In this case, disable the **Enable CAPI Synchronization** setting.

By default, this setting is disabled.

## Unprotected CAPI key storage?

The **Unprotected CAPI key storage?** setting specifies that when an Entrust Entelligence Security Provider user exports a profile to Microsoft CAPI (Cryptographic Application Programming Interface), they export unprotected keys. When this setting

is enabled, the user is not prompted to enter any further information during the export operation.

By default, this setting is disabled.

## Private key export from CAPI?

The **Private key export from CAPI?** setting allows you to specify whether Microsoft CAPI (Cryptographic Application Programming Interface) can export a user's keys to another application.

For some Entrust desktop applications, enabling this setting only takes effect if the **Enable CAPI Synchronization** setting is enabled (see [“Enable CAPI Synchronization” on page 422](#)). For more information on which applications this setting applies to, see your Entrust desktop application documentation.

By default, this setting is disabled.

## Number of key pairs

The **Number of key pairs** setting allows you to specify whether users have one or two key pairs in their Entrust profile.

When you change this setting for an existing user, the change does not take effect when the user next logs in to an Entrust desktop application. When changing this setting to 1 (that is, when changing a 2-key-pair user to a 1-key-pair user), there are several ways that this change can take effect in the user's profile:

- Recover the user's keys (see [“Recovering user key pairs” on page 162](#)).
- Initiate a DN change for the user (see [“Modifying distinguished names” on page 192](#)).
- Wait until the user's keys update automatically (see [“Configuring user properties” on page 219](#)).

Before changing the user from a 2-key-pair user to a 1-key-pair user, you can also revoke the user's encryption certificate or update the user's keys. Doing this ensures that the Entrust desktop application generates a new key pair for the user and requests a new dual-usage certificate from Security Manager. See [“Revoking user certificates” on page 174](#) and [“Recovering user key pairs” on page 162](#).

When changing this setting to 2 (when changing a 1-key-pair user to a 2-key-pair user), there are several ways that this change can take effect in the user's profile:

- Recover the user's keys (see [“Recovering user key pairs” on page 162](#)).
- Initiate a DN change for the user (see [“Modifying distinguished names” on page 192](#)).
- Wait until the user's keys update automatically (see [“Configuring user properties” on page 219](#)).

When you set a user as a 1-key-pair user, you must set the **Key type for signatures** policy setting to an RSA or EC value. Encryption is not supported with DSA. See [“Key type for signatures” on page 411](#).

If **Number of key pairs** is set to 1, you must disable **Enforce token usage**. The use of hardware tokens is not supported for 1-key-pair users. If you select this setting for a 1-key-pair user, the user cannot create an Entrust profile. See [“Enforce token usage” on page 413](#).

By default, this setting is set to 2.

See [“Creating user profiles” on page 158](#) for more details.

## Prevent single login register

The **Prevent single login register** setting allows you to prevent or allow users from registering an Entrust desktop application for Single Login.

When this setting is enabled, users are not allowed to manually register Entrust desktop applications through the Entrust Security Warning dialog box. All Entrust desktop applications must be automatically registered for those users. See also [“Delay single login register” on page 424](#).

By default, this setting is disabled.

## Delay single login register

The **Delay single login register** setting allows you to delay users from registering an Entrust desktop application for Single Login.

When this setting is enabled, users must wait five seconds before they can register the application for single login. This setting is useful if you want to ensure that users understand the implications of registering an application for Single Login before doing so. See also [“Prevent single login register” on page 424](#).

By default, this setting is disabled.

## Maximum bad login attempts

The **Maximum bad login attempts** setting allows you to configure the number of bad password attempts to allow before a suspension of a user profile is issued. If 0, the user has an unlimited number of bad password attempts.

By default, this setting is set to 0.

You can also configure the time window in which to record bad password attempts (see [“Login attempt window” on page 425](#)).

For more information about this setting, see the information about password management in the Entrust desktop application documentation.



## Login attempt window

The **Login attempt window** setting allows you to configure the time window in which to record bad password attempts as set in “[Maximum bad login attempts](#)” on [page 424](#).

By default, this setting is set to 1 minute.

For more information on this setting, see the information on password management in the Entrust desktop application documentation.

## ICE settings signed

The **ICE settings signed setting** requires that your Entrust/ICE Administrator settings are signed before you can use them.

By default, this setting is disabled.

## ICE settings ignored

The **ICE settings ignored** setting instructs Entrust/ICE to ignore any ICE Administrator settings.

By default, this setting is disabled.

## Enable the use of an ARL cache

The **Enable the use of an ARL cache** setting allows you to cache ARLs.

By default, this setting is enabled (ARLs are cached).

## Enable the use of a CRL cache

The **Enable the use of a CRL cache** setting allows you to cache CRLs.

By default, this setting is enabled (CRLs are cached).

## Enable the use of a XCert cache

The **Enable the use of a XCert cache** setting allows you to cache cross-certificates.

By default, this setting is enabled (cross-certificates are cached).

## Enable the use of a Cert cache

The **Enable the use of a Cert cache** setting allows you to cache user certificates.

By default, this setting is enabled (user certificates are cached).

## Secure Delivery SMTP

The **Secure Delivery SMTP** setting allows you to integrate users of Entrust Entelligence E-Mail Plugin 6.0 (formerly Entrust/Express) with the ZixIt SecureDelivery service. E-Mail Plugin requires the SMTP address of the ZixIt server in order to send mail to the server and find a certificate in the directory. Enter the SMTP address of the ZixIt server here.

By default, this setting is disabled (there is no server address listed).

## Content Scanner SMTP

The **Content Scanner SMTP** setting allows you to use content scanning software with Entrust Entelligence E-Mail Plugin (formerly Entrust/Express). You can configure E-Mail Plugin to decrypt all encrypted outbound email messages and scan them for content improprieties as dictated by the content policy of your organization. In order to do so, the outbound message is automatically encrypted for the email gateway and the gateway is added to the distribution list.

Each time an outbound message is copied to the default address, a window appears to inform the user that the message was copied to the address in question. The user can select a checkbox so that this window does not appear in the future.

In order to use this feature, your organization must purchase and set up a server-based content scanning application capable of handling encrypted messages.

By default, this setting is disabled.

For more information on this setting, see the information on content scanning in the Entrust desktop application documentation.

## Express Search Source Order

The **Express Search Source Order** setting allows you to define the default certificate search order. In Entrust Entelligence E-Mail Plugin (formerly Entrust/Express), the default certificate search order is:

- Certificate cache
- X.500 directory (Directory)
- Entrust Address Book (EAB)
- Exchange Global Address List (GAL)

The certificate cache is always searched first, but E-Mail Plugin lets you allow users to change the search order of the other three sources.

By default, this setting is `Directory EAB GAL`.

## Check e-mail on verification

When the **Check email on verification** setting is enabled, Entrust Entelligence E-Mail Plugin (formerly Entrust/Express) compares the email address in the signing certificate obtained from the received message against the email address for the user sending the message. If the email addresses do not match, the user receives a warning.

By default, this setting is disabled.

## Check e-mail on encryption

When the **Check e-mail on encryption** setting is enabled, Entrust Entelligence E-Mail Plugin (formerly Entrust/Express) checks the email address field in the encryption certificate obtained for the user and compares it against the email address that was used to find the certificate in the certificate repository. If the email addresses do not match, the user has the option to cancel sending the message.

By default, this setting is disabled.

## Cross Cert DNs

The **Cross Cert DNs** setting allows you to specify a list of DNs of cross-certified CAs whose cross certificates are included in e-mail messages. Each DN in the list is delimited by <>.

## Auto-Associate MS Outlook

The **Auto-Associate MS Outlook** setting allows you to automatically associate the certificates managed by Entrust Entelligence Security Provider with Microsoft Outlook. When this setting is enabled, the user does not have to set up an association manually in Microsoft Outlook.

By default, the automatic association is enabled.

## Searchbase Search Order

The **Searchbase Search Order** setting allows you to choose the searchbase sequence when client applications attempt to search for users in Security Manager. Enter the friendly names of the searchbases and separate each with a semi-colon. For example, to search through the Finance searchbase first, then the Development searchbase, and, finally, the HR searchbase, enter the following text:

Finance;Development;HR

Leave this setting blank to search through all searchbases.

## CRL grace period

The **CRL grace period** setting allows you to specify the amount of time (in hours) that a user can continue to use a CRL after it expires. This value is added to the time calculated based on the [“CRL grace percentage”](#) setting. You must set the value as an integer from 0 to 4.

By default, this setting is set to 2.

## CRL grace percentage

The **CRL grace percentage** setting allows you to specify the amount of time (as a percent of the CRL lifetime) that a user can continue to use a CRL after it expires. You must set this value as an integer from 0 to 100. If you set this setting to 0, the user cannot use the CRL after it expires. If you set this setting to 100, the user can still use the CRL after it expires, for an amount of time equivalent to its lifetime. See also the [“CRL grace period”](#) setting.

By default, this setting is set to 100.

## Reg. Pwd Max Fail

The **Reg. Pwd Max Fail** setting allows you to specify the number of times a user can enter a registration password incorrectly before being locked out. A value of 0 means that the user is never locked out.

This setting is used by Administration Services. It is not used by Security Manager Administration.

By default, this setting is set to 3.

## Reg. Client type

The **Reg. Client type** setting allows you to specify which client applications users should use to create their profiles through Administration Services. Use a space to separate values. The allowed values are:

- `entelligence`, to use the Entrust Entelligence profile creation wizard
- `direct`, to use the Extranet Profile Creation dialog
- `zf-local`, to use the Entrust TruePass Zero Footprint Local EPF applet (desktop profile)
- `zf-roaming`, to use the Entrust TruePass Zero Footprint Local EPF applet (roaming profile)
- `zf-msf`, to use the Entrust TruePass Zero Footprint Microsoft Windows Security Framework applet to create a digital ID stored in the computer's digital ID store or on a smart card

- **none**, to use the Administration Services registration service to create user profiles

This setting is used by Administration Services. It is not used for users created through Security Manager Administration.

By default, all client applications are permitted.

## Self-admin policy

The **Self-admin policy** setting allows you to specify how self-administration requests are handled by Administration Services. The choices are

- **execute**, or perform the request immediately
- **queue**, or queue the request for administrator approval
- **none**, or do not perform self-administration requests

This setting is used by Administration Services. It is not used by Security Manager Administration.

By default, this setting is set to execute.

## Public Token Certs

The **Public Token Certs** setting allows you to specify whether to write a user's encryption and verification certificates to the smart card's public storage. Normally, the user's certificates are written to the smart card's private storage. However, in some cases if you are creating Windows Smart Card Logon users, and those users log in to a Windows XP environment, you must enable this setting. Depending on the Enterprise desktop application in use, you may need to place the user's certificates in the smart card's public storage so that Windows XP applications can access them. When this setting is enabled, a hash of the certificates is stored in the private storage as well, where it cannot be overwritten.

By default, this setting is disabled.

## Enforce protected key transfer

The **Enforce protected key transfer** setting allows you to specify whether a user's keys are encrypted before they are transferred to a hardware token. The hardware token must support encryption of keys during transfer if this setting is enabled.

By default, this setting is disabled.

## Allow Spillover File for Tokens

If a token or smart card user's hardware device runs out of available memory, it cannot store all the user's keys during the user's next key update. The **Allow Spillover File for Tokens** setting specifies whether client applications store the oldest keys in an

APF file (such as `username.apf`) if the user's hardware device runs out of available memory. If enabled, client applications store the oldest keys in an APF file. If disabled, client applications delete the oldest keys.

By default, this setting is enabled (client applications store the oldest keys in an APF file).

## Messaging Server SMTP

The **Messaging Server SMTP** setting allows you to specify the email address of the Entrust Messaging Server that users of this policy contact when sending secure email. You must use this syntax:

```
any_name@Entrust_Messaging_Server_machine_name.your_domain
```

For example, if Messaging Server is installed on a machine called `EMS1.CompanyOne.com`, an appropriate email address would be:

```
Segue@EMS1.CompanyOne.com
```

The value must be a text string no longer than 1000 characters.

By default, this setting is blank.

## Symmetric Key Access

The **Symmetric Key Access** setting allows you to enable a client to access a user's private key so that it can decrypt data on behalf of the user without accessing the user's profile.

By default, this setting is disabled.

## Algorithm for profile protection

The **Algorithm for profile protection** setting allows you to specify the algorithm used to encrypt the Entrust profile that is created or recovered using Entrust Intelligence Desktop Solutions.

The options are CAST or Triple DES. You can change this setting during creation or recovery only.

If the user's policy certificate changes, existing profiles continue to use the original algorithm. In order to switch the Entrust profile to the newly configured protection algorithm, you need to set the user for key recovery and recover the user's Entrust profile.

By default, this setting is set to CAST.

## Allow Self Revocation

The **Allow Self Revocation** setting allows you to enable or disable this capability for end users. If it is enabled, Security Manager accepts CMP self revocation requests for a particular user.

The audit logs generated now indicate when an end user has performed a self-revocation.

By default, this setting is disabled.

# Certificate definition policy attributes reference

Certificate definition policies define the certificates associated with user types, so that the policy attributes you define apply to every user with the associated certificate. For example, when you define attributes for the encryption certificate definition, every user with the encryption certificate is governed by those policies.

This section describes all the policy attributes that you can configure in the certificate definition policies.

Security Manager provides default certificate definition policies to suit most users' requirements. All the settings apply to V2 key-pair users (see ["Configuring users' key pairs" on page 107](#)), but only some apply to V1 key-pair users. See the descriptions for each individual policy setting to see if it also applies to V1 key-pair users.

## Policy Certificate expires in (days)

The **Policy Certificate expires in (days)** setting specifies how long (in days) policy certificates are valid.

Each time a user logs in online to Security Manager, the user's policy certificate refreshes. If any changes are made to the policy certificate, these are picked up by the user when the user logs in.

You can set the **Policy Certificate expires in (days)** setting from 1 to 3650 days. By default, all policy certificates expire after 30 days.

This setting does not apply to V1-key-pair users.

## Certificate lifetime

The **Certificate lifetime** setting specifies how long (in months) a user certificate is valid.

You can set the **Certificate lifetime** setting from 2 to 120 months.

---

**Note:** It is possible to increase the certificate lifetime beyond the maximum of 120 months up to the lifetime of the current CA certificate through the certificate definition policy by customizing the `master.certspec` file.

---

Security Manager will use this setting unless one of the following conditions is met:

- If there is a setting defined for a user, Security Manager uses that value, unless the policy setting **Ignore per user lifetime** is enabled (see ["Ignore per user lifetime" on page 434](#)).



- If the certificate request from the client application includes a lifetime value, Security Manager uses that value unless the policy setting **Enable CMP override** is enabled (see [“Enable CMP override” on page 438](#)).
- If the policy setting **Certificate lifetime (Days)** is set to a value other than 0, Security Manager uses that setting (see [“Certificate lifetime \(Days\)” on page 433](#)).

This setting also applies to V1-key-pair users.

## Certificate lifetime (Days)

The **Certificate lifetime (Days)** setting specifies how long (in days) user certificates are valid.

You can set the **Certificate lifetime** setting from 3 to 3650 days. If 0, Security Manager uses the **Certificate lifetime** policy setting (see [“Certificate lifetime” on page 432](#)).

---

**Note:** It is possible to increase the certificate lifetime beyond the maximum of 3650 days up to the lifetime of the current CA certificate through the certificate definition policy by customizing the `master.certspec` file.

---

For values other than 0, Security Manager will use this setting unless one of the following conditions is met:

- If there is a setting defined for a user, Security Manager uses that value, unless the policy setting **Ignore per user lifetime** is enabled (see [“Ignore per user lifetime” on page 434](#)).
- If the certificate request from the client application includes a lifetime value, Security Manager uses that value unless the policy setting **Enable CMP override** is enabled (see [“Enable CMP override” on page 438](#)).

This setting also applies to V1-key-pair users.

## Private key usage period

The **Private key usage period** setting allows you to define how long the private keys associated with this policy should be used. This setting applies to all signing private keys defined by the certificate definition associated with this policy certificate.

You can set the **Private key usage period** setting as a percentage (1% to 100%) of the certificate lifetime. For example, if you define **Certificate lifetime** as 12 months, and you set **Private key usage period** to 50%, the private keys should be updated after 6 months.

Some Security Manager client applications use this setting to decide when to attempt key updates, but some client applications refuse to use keys when past the private key usage period.

This setting also applies to V1-key-pair users.

## Ignore per user lifetime

The **Ignore per user lifetime** setting allows you to set Security Manager to ignore per-user certificate lifetime values. For information about setting an individual user's certificate lifetime, see [“Configuring user key update options” on page 230](#). In fact, you can control the certificate lifetime through a number of different settings. For more information, see [“Certificate lifetime” on page 432](#) and [“Certificate lifetime \(Days\)” on page 433](#).

This setting also applies to V1-key-pair users.

## Publishing policy

The **Publishing policy** setting allows you to determine which certificates Security Manager publishes to the directory in response to a directory restore or key management operation—all the certificates, only the latest certificates, or none.

You can also control certificate publishing for the latest certificates by the certificate request message from the client. If you select **Use CMP publish flag**, Security Manager checks the publish flag in the client request message first. If the CMP publish flag is false, Security Manager does not publish the certificate, regardless of the value of this **Publishing policy** setting. If **Use CMP publish flag** is disabled, this **Publishing policy** setting controls certificate publishing. (See [“Use CMP publish flag” on page 439](#).)

Publishing all the certificates can cause problems with Entrust desktop applications. Similarly, publishing no encryption certificates at all can cause problems. It is best, whenever possible, to publish only the latest encryption or dual-usage certificate, and to publish no verification certificates.

You can set the **Publishing policy** setting to `all`, `latest`, or `none`.

---

**Note:** The publishing policy applies only to the current certificate type. If the user's certificate type changes, certificates from the old certificate type are no longer published when a certificate is updated or a directory restore operation is done.

---

This setting also applies to V1-key-pair users.

## Publish revoked certs.

The **Publish revoked certs.** setting allows you to determine whether revoked certificates are published to the directory.

Publishing multiple certificates can cause problems with Entrust desktop applications. It is best, whenever possible, not to publish revoked certificates.

If you select this setting, Security Manager publishes no certificates under the following circumstances:

- the **Publishing policy** is to publish only the latest certificates (see [“Publishing policy” on page 434](#))
- the latest certificate is revoked

This setting also applies to V1-key-pair users.

## Publish expired certs.

The **Publish expired certs.** setting allows you to determine whether expired certificates are published to the directory. (The expired status applies only to the certificate, not to any of the signing certificates.)

Publishing multiple certificates can cause problems with Entrust desktop applications. It is best, whenever possible, not to publish expired certificates.

If you select this setting, Security Manager publishes no certificates under the following circumstances:

- the **Publishing policy** is to publish only the latest certificates
- the latest certificate is expired

This setting also applies to V1-key-pair users.

## Publishing DN

The **Publishing DN** setting allows you to determine which certificates are published if a user's DN changes and the certificate's history is published.

You can set the **Publishing DN** setting to

- `current` to cause Security Manager to publish certificates to the user's current DN
- `match` to cause Security Manager to publish certificates to the user's current DN only if the subject of the DN matches the current DN

This setting applies to V1-key-pair users only if the setting is set to `match` and the subject of the DN matches the current DN. See [“Publishing DN” on page 435](#).

## Publish at key update

The **Publish at key update** setting allows you to determine whether Security Manager publishes a new certificate to the directory when the administrator updates a user's keys. To enable publishing of the certificate, the following conditions must be met:

- **Publishing policy** must be set to **all** or **latest**
- Security Manager must generate the keys rather than an Entrust client
- Key backup must be enabled

This setting does not apply to V1-key-pair users.

## Exclude privateKeyUsagePeriod

The **Exclude privateKeyUsagePeriod** setting allows you to exclude the `privateKeyUsagePeriod` extension from certificates during certificate generation.

This setting overrides the value of the `noPrivateKeyUsage` field in the [Advanced Settings] section of the `master.certspec` file. (See [“Excluding certificate extensions by certificate type” on page 593.](#))

The Security Manager Control Command Shell global advanced setting `userCertPrivateKeyUse` overrides all the other settings related to this extension when it is set to exclude the extension. When it is set for inclusion of the extension, it does not override the other settings. (See the *Security Manager Operations Guide*.)

This setting also applies to V1-key-pair users.

## Exclude basicConstraints

The **Exclude basicConstraints** setting allows you to exclude the `basicConstraints` extension from certificates during certificate generation.

This setting overrides the value of the `noBasicConstraints` field in the [Advanced Settings] section of the `master.certspec` file. (See [“Excluding certificate extensions by certificate type” on page 593.](#))

The Security Manager Control Command Shell global advanced setting `userCertBasicConst` overrides all the other settings related to this extension when it is set to exclude the extension. When it is set for inclusion of the extension, it does not override the other settings. (See the *Security Manager Operations Guide*.) This setting also applies to V1-key-pair users.

## Exclude CDP

The **Exclude CDP** setting allows you to exclude the CDP extension from certificates during certificate generation.

This setting overrides the value of the `noCRLDistPoints` field in the [Advanced Settings] section of the `master.certspec` file. (See [“Excluding certificate extensions by certificate type” on page 593.](#))

This setting also applies to V1-key-pair users.

## Exclude entrustVersInfo

The **Exclude entrustVersInfo** setting allows you to exclude the `entrustVersInfo` extension from certificates during certificate generation.

This setting overrides the value of the `noEntrustVersInfo` field in the [Advanced Settings] section of the `master.certspec` file. (See [“Excluding certificate extensions by certificate type” on page 593.](#))

This setting also applies to V1-key-pair users.

## Exclude subjectKeyId

The **Exclude subjectKeyId** setting allows you to exclude the `subjectKeyId` extension from certificates during certificate generation.

This setting overrides the value of the `noSubjectKeyId` field in the [Advanced Settings] section of the `master.certspec` file. (See [“Excluding certificate extensions by certificate type” on page 593.](#))

This setting also applies to V1-key-pair users.

## Exclude subjectAltName

The **Exclude subjectAltName** setting allows you to exclude the `subjectAltName` extension from certificates during certificate generation.

## Exclude certificatePolicy

The **Exclude certificatePolicy** setting allows you to exclude the `certificatePolicy` extension from certificates during certificate generation.

If you do not select **Exclude certificatePolicy**, Security Manager places encryption OIDs and the encryption lifetime in encryption certificates, and verification OIDs and the verification lifetime in verification certificates. Security Manager determines the `certificatePolicy` values to use as follows:

- if there are per-user `certificatePolicy` OIDs, they take precedence
- if there are no per-user OIDs, the `certificatePolicy` extension from the `master.certspec` file is used
- if there is no `certificatePolicy` extension in the `master.certspec` file, the `certificatePolicy` extension from the Entrust client certificate request message is used
- otherwise, the `certificatePolicy` OIDs specified in the security policy are used

This setting also applies to V1-key-pair users.

## Exclude subjectAltName parts

Allows you to exclude particular `subjectAltname` components. Valid names are:

- NONE

- email
- dns
- dn
- uri
- ip
- oid
- upn
- other
- edi
- x400
- MsGUID
- FASC-N
- Permlid

For more information, see [“Excluding the subjectAltName from certificate definitions” on page 270](#).

This setting does not apply to V1-key-pair users.

## SubjectAltName criticality

Sets the `subjectAltname` extension to critical for a particular certificate definition policy. For more information, see [“Setting the criticality of the subjectAltName extension” on page 273](#).

This setting does not apply to V1-key-pair users.

## Enable CMP override

The **Enable CMP override** setting allows you to determine whether Security Manager uses or ignores any policy settings provided in the certificate request message from the Entrust client. The message can include the following policy settings:

- certificate lifetime
- key type (or algorithm) and size (for client-generated keys)
- key backup
- publish flag
- `privateKeyUsagePeriod` extension
- key usage extension (key usage policy)

In the case of the certificate lifetime, per-user settings override the client message settings (see [“Certificate lifetime” on page 432](#) and [“Certificate lifetime \(Days\)” on](#)

[page 433](#)). If there are values for key size, key type (or algorithm), and key backup in the client message, Security Manager does not validate them, regardless of the setting for **Enforce client policy** (see [“Enforce client policy” on page 439](#)).

This setting does not apply to V1-key-pair users.

## Allow unknown extensions

The **Allow unknown extensions** setting allows you to determine whether to include in the certificate any unknown extensions in the client request message.

This setting does not apply to V1-key-pair users.

## Enforce client policy

The **Enforce client policy** setting allows you to determine whether Security Manager validates the policy values provided in the client message. These policy values include:

- key type (or algorithm)
- key backup
- key usage policy

This setting does not apply to V1-key-pair users.

## Only latest key can sign CMP

The **Only latest key can sign CMP** setting allows you to determine whether Security Manager should ensure that the private key that signed the client request is the latest key. (Certificate Management Protocol (CMP) is the protocol used for communication between Entrust clients and Security Manager.) This setting applies only to certificates that are allowed to sign client request messages.

This setting does not apply to V1-key-pair users.

## Key can sign CMP

The **Key can sign CMP** setting allows you to determine whether the client can use the private key for signing request messages. Automatic key update is not possible unless one of the verification keys has this setting enabled.

This setting does not apply to V1-key-pair users.

## Use CMP publish flag

The **Use CMP publish flag** setting allows you to determine whether Security Manager publishes certificates according to the publish flag in the client request message. If you select this setting, Security Manager publishes a certificate if the CMP publish flag is true, regardless of the value of the **Publishing policy** setting. Similarly,

Security Manager does not publish a certificate if you select this setting and the CMP publish flag is false.

If you do not select this setting, Security Manager determines whether to publish the certificate according to the **Publishing policy** value. (See [“Publishing policy” on page 434.](#))

This setting also applies to V1-key-pair users.

## Algorithm for key pair

The **Algorithm for key pair** setting specifies the type of user keys to be generated for V2 client applications. When you configure this setting, it is important to consider which key types the client application supports. For example, if you specify DSA-1024 as the key type, and the client application does not support DSA-1024, key management will fail.

---

**Note:** When user encryption keys are server-generated (see [“Key usage policy” on page 441](#) and [“Generate key at client” on page 441](#)) and the **Algorithm for key pair** setting is an RSA value, then it is superseded by the server-generated RSA user encryption keys set in the security policy (see the `policy userEncryptionAlg` command in the *Security Manager Operations Guide*).

---

You can set **Algorithm for key pair** to one of the following values (using uppercase letters only):

- RSA-1024
- RSA-2048
- RSA-3072
- RSA-4096
- RSA-6144
- DSA-1024
- ECDSA-192

This value is supported for backwards compatibility. It is the same as EC-P-192.

- EC-<curve>

Where <curve> is a named elliptic curve. For a list of supported elliptic curves, see the *Security Manager Operations Guide*.

This setting does not apply to V1-key-pair users.



## Back up private key

In the case of client-generated keys, the **Back up private key** setting allows you to specify whether the client should send the key to Security Manager for backup. Keys generated by Security Manager are always backed up.

Verification keys cannot be backed up. If you specify a **Key usage policy** of **verification**, you must make sure that **Back up private key** is disabled.

You must back up keys generated by Security Manager, so if **Generate key at client** is disabled, you must make sure that you enable **Back up private key**.

This setting does not apply to V1-key-pair users.

## Generate key at client

The **Generate key at client** setting allows you to specify whether the client generates the key.

The client must generate verification keys. If you specify a **Key usage policy** of **verification**, you must make sure that **Generate key at client** is enabled.

This setting does not apply to V1-key-pair users.

## Key usage policy

The **Key usage policy** setting allows you to specify the purpose of the client-generated key pair as encryption, verification, or both. This setting is typically required for hardware devices that need to know what the key is to be used for before creating it.

If **Enable CMP override** is disabled, this setting is mandatory. If **Enable CMP override** is enabled, you do not need to specify this value, but if you do set this value, it must match the `keyUsage` extension in the client request. (See [“Enable CMP override” on page 438](#).) The **Key usage policy** value must also match the value set in the `master.certspec` file. If it does not, an error occurs.

If there is a `keyUsage` extension value in the client request message, that value must match any value defined in the `master.certspec` file. Otherwise, an error occurs.

Security Manager looks for the key usage value to use as follows:

- if there is a value in the `master.certspec` file, Security Manager uses this value
- if not, Security Manager uses the value from the client message
- otherwise, Security Manager uses this **Key usage policy** setting

If there is more than one value defined, the values must match. Otherwise, an error occurs.

This setting does not apply to V1-key-pair users.

## CSP to manage keys

The **CSP to manage keys** setting allows you to specify the Cryptographic Service Provider (CSP) in cases in which the CSP used by the Entrust client manages the user's keys.

---

**Note:** If the **CSP to manage keys** setting is modified in an existing user policy, the new value applies to all newly generated keys. If a user is recovered, it applies to all keys—existing keys and new keys. If the certificate type of a user changes, existing keys remain in the CSP specified in the policy of the certificate definition settings of their original certificate type, while new keys are stored in the CSP specified in the certificate definition settings of the new certificate type.

---

This setting does not apply to V1-key-pair users.

## Protect key storage for CSP

The **Protect key storage for CSP** setting allows you to specify the level of key storage protection for keys stored in a CSP key storage medium. The levels of protection are:

- password protection
- notification
- other

This setting does not apply to CSPs that are password-protected by default, such as all Entrust CSPs and most smart card CSPs.

By default, the **Protect key storage for CSP** setting is enabled, and password-protects the CSP key storage medium.

This setting does not apply to V1-key-pair users.

## Private key export from CSP

The **Private key export from CSP** setting allows you to specify whether to allow export of the user's private keys from the CSP.

This setting does not apply to V1-key-pair users.

## Symmetric Key Access

The **Symmetric Key Access** setting allows you to enable a client to access a user's private key so that it can decrypt data on behalf of the user without accessing the user's profile.

By default, this setting is disabled.

## Force client key generation in CSP

The **Force client key generation in CSP** setting creates a client-generated key pair in the destination CSP if it is not already backed up in the CSP.

When this setting is false (or not selected), the application (Entrust Entelligence Security Provider for Windows for example) generates the key pair in memory, and then imports it into the destination CSP. This setting only affects encryption and dual-usage key pairs. Security Provider for Windows always generates signing key pairs in the destination CSP (never in memory) so this setting has no impact.

See the *Entrust Entelligence Security Provider for Windows Administration Guide* for more information about CSPs and this policy.

This setting does not apply to V1-key-pair users.

## Update cert. at % of lifetime

The **Update cert. at % of lifetime** setting allows you to define the time when the user's certificates should be updated. With this setting, you specify the time as a percentage of the certificate's lifetime between 0 and 99%. If you define this setting, the client checks the certificate lifetime of the latest encryption certificates and the private key lifetime of the latest verification certificates, and updates the certificates when the identified percentage of the lifetimes remains.

If you set **Update cert. at % of lifetime** to 0, the certificate updates at 50% of the certificate's lifetime, or 100 days, whichever is less (closer to the expiry date).

See [“Cert. update date”](#) setting for additional information.

This setting does not apply to V1-key-pair users.

## Enable cert. update date

The **Enable cert. update date** setting allows you to enable or disable the date set for **Cert. update date**. (See [“Cert. update date”](#) on page 443.)

By default, the **Enable cert. update date** setting is disabled.

This setting does not apply to V1-key-pair users.

## Cert. update date

The **Cert. update date** setting allows you to define a date after which the client should request a new certificate.

Certificate and key updates for V2-key-pair users are performed according to the certificate definition policy, where the [“Update cert. at % of lifetime”](#) setting specifies when the associated key pair updates. Updates are also performed when:

- the **Update cert. at % of lifetime** setting is approaching expiry

- the **Cert. update date** is reached. If this value is not specified, the **Update cert. at % of lifetime** setting is used.

You can set **Cert. update date** to any date and time value. For example, you could enter the date 12/03/2002 7:00:00 PM to indicate a date of 7:00 PM on December 3, 2002. The client ignores any date in the future.

This setting does not apply to V1-key-pair users.

## Certificate Signing Alg

The **Certificate Signing Alg** setting specifies the algorithm that Security Manager uses when signing certificates.

You can set the **Certificate Signing Alg** setting to one of the following values (using uppercase letters only): DEFAULT, RSA-SHA1, RSA-SHA224, RSA-SHA256, RSA-SHA384, RSA-SHA512, RSAPSS-SHA1, RSAPSS-SHA224, RSAPSS-SHA256, RSAPSS-SHA384, RSAPSS-SHA512, ECDSA-SHA1, ECDSA-SHA224, ECDSA-SHA256, ECDSA-SHA384, or ECDSA-SHA512.

If DEFAULT, Security Manager uses the current CA signing algorithm defined in the Security Manager security policy. Master Users can view or change the current CA signing algorithm by running the `policy signatureAlg` command in the Security Manager Control Command Shell. See the *Security Manager Operations Guide* for details.

If you specify an algorithm other than DEFAULT, you must select an algorithm that matches the key type of your CA. For example, if your CA has an RSA key type, you must select an RSA signing algorithm. If you select an algorithm with a different key type, errors will occur when Security Manager attempts to sign certificates.

---

**Note:** It is recommended that you do not select an algorithm that is weaker than the CA signing algorithm. For example, if the CA signing algorithm is RSA-SHA256, do not select RSA-SHA1. Client applications that use certificates must verify the CA certificate and must be compatible with the CA signing algorithm.

---

This setting also applies to V1-key-pair users.

## Revoke superseded certs.

The **Revoke superseded certs.** setting controls whether Security Manager automatically revokes a certificate when it is superseded by a new certificate.

When sending new certificates to the user's client application, if the certificate definition policy for the new certificate has the **Revoke superseded certs.** setting enabled, Security Manager marks any older certificates in the certificate stream for automatic revocation. A certificate stream for a user is identified by a certificate type

and certificate definition. A user can have more than one stream if the user ever changed certificate types.

Security Manager will revoke older certificates in the certificate stream when it receives an acknowledgement message from the client application. If the client application fails to send an acknowledgement message, Security Manager will revoke only the new certificate.

When revoking older certificates in a certificate stream, Security Manager will revoke each certificate with the “Superseded” reason (see [“Reasons for revoking certificates” on page 174](#)).

Certificate synchronization is possible with V2 client applications. Certificate synchronization allows multiple copies of the same digital ID to be synchronized, even if a particular copy of the digital ID does not have the latest verification certificate. Enabling the **Revoke superseded certs.** setting can cause certificate synchronization to fail, if a particular copy of a user’s digital ID has only the older verification certificates that were automatically revoked by Security Manager.

It is recommended that you do not enable the **Revoke superseded certs.** setting if you enabled **Always issue a new CRL after certificate revocation** in the Security Policy (see [“Configuring the Administration Policy” on page 85](#)). When the **Always issue a new CRL after certificate revocation** Security Policy setting is enabled, Security Manager will automatically update the applicable CRL immediately after a certificate is revoked. If different Security Manager processes attempt to revoke certificates around the same time, timeout errors may occur due to locked processes.

If you enable the **Revoke superseded certs.** setting, CRLs can greatly increase in size. Large CRLs—particularly large combined CRLs—can decrease the performance of Security Manager, and some client applications have problems validating very large CRLs. The following factors can increase the size of CRLs:

- large number of users
- short certificate lifetimes
- using combined CRLs
- retaining expired certificates on CRLs (see [“Configuring the Administration Policy” on page 85](#))
- `DisableCombExpCheck=1` in the `entmgr.ini` file (see the *Security Manager Operations Guide*)

By default, the **Revoke superseded certs.** setting is disabled.

This setting also applies to V1-key-pair users. It is not supported for Proto-PKIX client applications.



# Modifying the user template file and user types

The **New User** dialog box lets Entrust PKI administrators with sufficient permissions add new users to Security Manager. This chapter provides you with instructions on modifying the **Naming** tab in this dialog box as required for your organization.

This chapter contains the following sections:

- [“Overview of user template file and user types” on page 448](#)
- [“Exporting the template definition file” on page 451](#)
- [“Customizing the New User dialog box” on page 452](#)
- [“Importing the template definition file” on page 461](#)
- [“Testing the template definition file” on page 462](#)

# Overview of user template file and user types

The user template file allows you to configure how Security Manager adds different types of users. To add other user types or modify existing ones, modify the user template definition file, called `usertype.templates`.

Topics in this section:

- [“Purpose of user types” on page 448](#)
- [“Available user types” on page 449](#)
- [“Summary of steps” on page 450](#)

## Purpose of user types

User types customize the **Naming** tab in the **New Users** dialog box and in the Directory Browser's **New Entry** dialog box.

User types allow you to:

- Choose what information goes into the CN portion of the user's DN. For example, if you are using the Person user type, the DN would be:  
`cn=FirstName LastName,ou=CA1,dc=entrust,dc=com`  
If you are using the Web server user type, the DN would be:  
`cn=hostname.domain.com,ou=CA1,dc=entrust,dc=com`
- Set various attributes and object classes in the directory when the new user is created.  
You may want to do this when auto-populating the certificate extension using directory information. For example, by default, if you enter an email address when creating a user using the Person user type, the email address is automatically populated into the certificate's extension. For more information, see [“Using the subjectAltName extension” on page 250](#).
- Restrict administrators to administer users with particular user types.

Changing the user types changes the structure of the user's directory entry.



## Available user types

The template definition file includes five user types:

- Person

The default user type for adding end users to Security Manager. By default, it allows you to specify a first and last name, multiple email addresses, and serial number. The DN created for the user consists of the first and last name.

- Web server

The default user type for adding servers to Security Manager. By default, it allows you to specify the server name and an optional description. The DN created for the server consists of the server name.

- Organizational Unit

You can select this user type in the **New Entry** dialog box in the Directory Browser application only. By default, you do not see Organizational Unit in the **New User** dialog box in Security Manager Administration. This user type is used in the creation of searchbases. For more information, see [“Administering searchbases” on page 341](#).

- Hardware

Activate this user type to add hardware devices to the Security Manager system.

- Person 4.0 PKI

Activate this user type if you want to include a user’s email address in the user’s DN. This user type is included to allow Entrust PKI administrators performing a Change DN operation to keep the user’s email address in the DN

It is recommended that you do not include the user’s email address in the user’s DN.

You can activate the Hardware and Person 4.0 PKI user types so that they appear in the **New User** dialog box. To activate either user type, see [“Activating Hardware and Person 4.0 PKI user types” on page 459](#).

---

**Note:** When you use Microsoft Active Directory, some of the end user types in the **New User** dialog box are named differently. For example, the default user type is **User** rather than **Person**. For more information, check the Entrust TrustedCare Online Web site (<https://www.entrust.com/trustedcare>) for a listing of related white papers and where to find them.

---

## Summary of steps

A Security Officer or Entrust PKI administrator with sufficient permissions is responsible for exporting, editing, and importing the template definition. You can use any text editor to edit the file.

---

**Attention:** It is recommended that you keep a secure backup of the template definition file. If you accidentally delete or misplace the backup, you can retrieve the file in its original form from your Security Manager downloaded files.

---

### To edit the template definition file

- 1** Export the file from Security Manager Administration to a location on your disk. See [“Exporting the template definition file” on page 451](#).
- 2** Open the file in a text editor.
- 3** Add or activate a user type:
  - To add a new user type from scratch, you add a user type and a new user type section. See [“To add a new user type” on page 452](#).
  - To activate the Hardware or Person 4.0 User user types, you add a new line of text in the user type section. See [“To activate the Hardware and Person 4.0 PKI user types” on page 459](#).
- 4** Add attributes to the new user type section. See [“Adding attributes to user types” on page 455](#).
- 5** Save the file using a new filename.
- 6** Import the file into Security Manager Administration. See [“To import the template definition file” on page 461](#).
- 7** Test the file. See [“To test the template definition file” on page 462](#).

Make sure you correctly format the information you add in the template definition file. Model your new user types and attributes after existing user types and attributes to ensure that you do not omit any vital information.

# Exporting the template definition file

If you are a Security Officer or an Entrust PKI administrator with sufficient permissions, you can export the template definition file to a location on your computer's disk. This is the first step in the process of adding or activating a user type.

---

**Note:** It is strongly recommended that you export a backup of the original `usertype.templates` file and store it in a safe location. This way you can restore from the backup if you encounter problems.

---

## To export the template definition file

- 1 Log in to Security Manager Administration. See [“Logging in to Security Manager Administration” on page 46](#).

Security Manager Administration appears.

- 2 Click **File > User Templates > Export**.

The Save As dialog box appears.

- 3 In the Save As dialog box, specify a filename and location for the `usertype.templates` file and click **Save**.

When you export the file, you should rename it in order to maintain a complete file history. If you do not rename the file, it is overwritten. Export the most recent file and store the backups on disk.

A dialog box appears to indicate successful export of the file.

- 4 Click **OK**.

You have now exported the file for editing. When you finish editing it, import the file as described in [“Importing the template definition file” on page 461](#).

# Customizing the New User dialog box

Users types refer to all people and devices that you add to Security Manager and are defined by the directory schema.

Topics in this section:

- [“Adding a new user type” on page 452](#)
- [“Adding attributes to user types” on page 455](#)
- [“Determining the structure of the common name for the Person user type” on page 459](#)
- [“Activating Hardware and Person 4.0 PKI user types” on page 459](#)

---

**Note:** Your directory schema must validate all new user types. You cannot add arbitrary or unsupported user types. To determine which user types your directory supports, consult a Directory Administrator or Security Officer. For general information about directories, see the *Security Manager Directory Configuration Guide*.

---

## Adding a new user type

Adding a new user type is a two-step process:

- First add the new user type to the User Type Template List (see [“To add a new user type” on page 452](#)).
- Then, create a new section for the new user type (see [“To add a new user type section” on page 453](#)).

Use the following instructions to add a new user type.

---

**Note:** The working section of the default `usertype.templates` file follows a series of instructions. All text on a line that begins with a semicolon (;) is instructional. Begin editing after the line reading `End of Instructions`.

---

### To add a new user type

- 1 Open the template definition file in a text editor.
- 2 After the instructions section, under the heading `[User Type Template List]` increase the `count=` value by 1.

For example, to add a new user type—in this example, `Internet User`—change `count=3` to `count=4` (as `Internet User` is the fourth user type in this file).

`[User Type Template List]`

```
count=4
0=Person
1=Web Server
2=Organizational Unit
```

The user type that appears first in the New User dialog box user type list is that which is assigned the 0 value in the [User Type Template List]. By default, this is the Person user type (0=Person).

You add the Organizational Unit user type to searchbases. It is available only in the New Entry dialog box in the Directory Browser (you do not see this user type in the **New User** dialog box).

- 3** Still under the heading [User Type Template List], add the new user type to the numbered list.

For example, add 3=Internet User below 2=Organizational Unit

```
[User Type Template List]
count=4
0=Person
1=Web Server
2=Organizational Unit
3=Internet User
```

Type the name of the new user type in the list below the last line. Use common English. You can include spaces and uppercase and lowercase letters.

You have now completed the first step in adding a new user type. To continue, see [“To add a new user type section” on page 453](#).

## To add a new user type section

- 1** Scroll to the bottom line (below the [Hardware] section) and type the section heading.

Make sure you enter between two square brackets [ ], the exact text that appears after the = sign in the user type you created in the previous procedure. For example:

```
[Internet User]
```

Type the name of the new user type in the list below the last line. Use common English. You can include spaces and uppercase and lowercase letters.

- 2** Below the heading, type the `id` number for the new user type.

This is the number assigned to the user type in the User Type Template List. The number in this example is 3.

```
[Internet User]
id=3
```

You can insert the new user type section anywhere in the file as long as the `id` number is correct.

**3** Type `count=` and the number of attributes you want to add to this user type.

Attributes refer to common user information, such as name, serial number, telephone number, and email address. At this point, you should know how many and what type of attributes you want to add for the new user type.

If you plan to add one attribute for the new user type, type `count=1`. In this example, two attributes are added to the user type:

```
[Internet User]
id=3
count=2
```

**4** Type `Structural Object Class=` and the name that represents this user.

You must know how the user type is represented in the directory. The names are generally not in plain English. Using our example, Internet User is represented in the directory as `inetOrgPerson`. Note the lowercase and uppercase letters.

```
[Internet User]
id=3
count=2
Structural Object Class=inetOrgPerson
```

Type the proper structural object class definition for the user type. You must type the class name as it is known in the directory. Ask a Directory Administrator if you need assistance.

**5** This step is optional. If you want the new user type to be available only using the Directory Browser's New Entry dialog box, add `DirBrowserOnly=1` below `Structural Object Class` (as described in [Step 4](#)).

If you use this setting, the new user type does not appear in the New User dialog box in Security Manager Administration. By default, this line only appears in the Organizational Unit type.

```
[Internet User]
id=3
count=2
Structural Object Class=inetOrgPerson
DirBrowserOnly=1
```

If you later decide that you want to include the new user type in the Security Manager Administration New User dialog box (where you add users to Entrust), simply remove the `DirBrowserOnly=1` line or change the 1 to 0.

**6** Type `description=` and a description of the user type.

The description appears in the New User dialog box and helps Entrust PKI administrators understand the user type. The example used here is very short. Do not hesitate to add more descriptive, conversational text if you think Entrust PKI administrators can benefit from such a description.

```
[Internet User]
id=3
count=2
Structural Object Class=inetOrgPerson
description=Internet user
```

You now add attributes for the user type. See [“To add an attribute for a new user type” on page 455](#).

---

**Note:** If you create a new user type and decide to remove the user type later, ensure that the user type you are deleting is not currently in use by any users. If a user exists and its type is deleted, then you may have problems displaying information related to that particular user in Security Manager Administration.

---

## Adding attributes to user types

You can add as many different attributes for each user type as you want. There are many attribute types in the directory. Use standardized attributes to ensure interoperability.

### To add an attribute for a new user type

**1** Open the `usertype.templates` file in a text editor and complete the section, [“Adding a new user type” on page 452](#).

**2** Under `description=Internet user` (in this example), type `0=`. Attributes are listed, in order, beginning with `0`.

```
[Internet User]
id=3
count=2
Structural Object Class=inetOrgPerson
description=Internet user
0=
```

Type `1=` for the next attribute after this one.

**3** Next, type the common name of the attribute, followed by a comma. This is what appears as the field label in the New User dialog box.

```
[Internet User]
id=3
count=2
Structural Object Class=inetOrgPerson
description=Internet user
0=Common Name
```

- 4** Next, type the attribute name as it is recognized in directory language. You must know the name of the attribute used in the directory. Using our example, `Common Name` is represented in the directory as `cn`. Then type a comma.

```
[Internet User]
id=3
count=2
Structural Object Class=inetOrgPerson
description=Internet user
0=Common Name,cn,
```

- 5** Next, type a value to determine whether and how the attribute information is added to the DN. Then type a comma.

Value	State of In DN checkbox in New User dialog box	Adds information in the attribute field to the DN? Yes/No.
0	Enabled and unchecked	Yes, if check box checked. No, if unchecked.
1	Disabled and checked	Yes. The attribute is automatically added in the DN.
2	Disabled and unchecked	No. The attribute cannot be added in the DN.
3	Enabled and checked	Yes, if check box checked. No, if unchecked.

```
[Internet User]
id=3
count=2
Structural Object Class=inetOrgPerson
description=Internet user
0=Common Name,cn,1,
```



- 6 Next, type 0 or 1 to make the attribute optional or mandatory in the directory. Type 0 to make this attribute optional in the directory, and 1 to make it mandatory. Then type a comma.

```
[Internet User]
id=3
count=2
Structural Object Class=inetOrgPerson
description=Internet user
0=Common Name,cn,1,0,
```

- 7 Next, type 1 to make the value you enter for this attribute unique across all available searchbases in the directory. Type 0 if you do not want to make this attribute unique. Then type a comma.

For example, if you make the common name value unique, you cannot add two users to Security Manager with the same common name.

```
[Internet User]
id=3
count=2
Structural Object Class=inetOrgPerson
description=Internet user
0=Common Name,cn,1,0,1,
```

---

**Note:** When you add a user with a unique attribute value, Entrust searches the available searchbases in the directory for this value to see if it currently exists. If this value is not found, you can use it. However, there is no way to guard against its reuse if it is later removed or changed using a third-party tool.

---

- 8 Optionally, add an auxiliary object class.

To do so, type the object class name and then a comma. The auxiliary object class is added to the entry before the attribute is added.

---

**Attention:** You cannot add auxiliary classes when using Microsoft Active Directory Application Mode (ADAM) or Active Directory Lightweight Directory Services (AD LDS). They are not supported.

---

For example:

```
[Internet User]
id=3
count=2
```

```
Structural Object Class=inetOrgPerson
description=Internet user
0=Common Name,cn,1,0,0,uniquelyIdentifiedUser
```

- 9 Optionally, indicate whether you want to enter multiple values for this attribute when creating users.

Use this option if you are configuring a `mail` attribute in your template, and you want to enter multiple email addresses. Enter `1` if you want to enter multiple values for the attribute, otherwise enter `0`. When entering multiple values during user creation, separate each with a space.

For example, you would use `0` for `Common Name`:

```
[Internet User]
id=3
count=2
Structural Object Class=inetOrgPerson
description=Internet user
0=Common Name,cn,1,0,1,uniquelyIdentifiedUser,0
```

---

**Note:** If you choose not to add an auxiliary class, you must add an extra comma before specifying this value. For example, `0=First Name,cn,1,0,0,,1`

---

If you were adding a `mail` attribute and want to add multiple email addresses, your entry appear as follows:

```
0=Email,mail,2,0,0,rfc822MailUser,1
```

You have now added an attribute to the new user type.

- 10 Repeat this procedure to add other attributes, increasing the `count=` for each one. For our example, we must add another attribute, since the `surname` attribute is a mandatory attribute for `iNetOrgPerson`.

```
[Internet User]
id=3
count=2
Structural Object Class=inetOrgPerson
description=Internet user
0=Common Name,cn,1,0,1,uniquelyIdentifiedUser,0
1=Surname,sn,2,1,0,,0
```

You have now added a new user type section to the template definition file.

- 11 Save the file using a new filename and import the file as described in [“To import the template definition file” on page 461](#).

## Determining the structure of the common name for the Person user type

In the template definition file, the Person user type includes an additional line in the Person section:

```
overrideCommonNameFormat=1
```

This override specifies how the cn (common name) value is formed (that is, showing the first name followed by the last name, or the other way around).

To form the cn as follows, enter 1:

```
"cn=First Name Last Name"
```

To form the cn as follows, enter 2:

```
cn="Last Name, First Name"
```

If the override value is undefined, missing, or a number other than 1 or 2 is entered, the cn consists of the value entered in the First Name field.

---

**Note:** A cn comprises two names, but the label associated with each is irrelevant. If you add a customized user type, it may not be the First and Last name that you are expecting. It could be two others, such as "Given Name" or "Family Name." It depends on the user type you are adding.

---

## Activating Hardware and Person 4.0 PKI user types

Complete this procedure to activate the Hardware and Person 4.0 PKI user types. See ["Available user types" on page 449](#) for information on these types.

### To activate the Hardware and Person 4.0 PKI user types

- 1 Open the template definition file in a text editor.
- 2 After the instructions section, under the heading [User Type Template List] increase the count= value by one.

For example, to activate one of the user types, change count=3 to count=4.

```
[User Type Template List]
```

```
count=4
```

```
0=Person
```

```
1=Web Server
```

```
2=Organizational Unit
```

- 3 Still under the heading [User Type Template List] add the Hardware or Entrust 4.0 PKI user type to the numbered list.

For example, add 3=Hardware below 2=Organizational Unit.

```
[User Type Template List]
count=4
0=Person
1=Web Server
2=Organizational Unit
3=Hardware
```

- 4 Under the appropriate user type section, update the id to use the next available number.

For example, set id=3 below [Hardware]:

```
[Hardware]
id=3
count=3
Structural Object Class=device
...
```

- 5 Save the file using a new filename and import the file as described in [“To import the template definition file” on page 461](#).

Sections for both the Hardware and Person 4.0 User user types are included in the `usertype.templates` file. This is all you need to do to add either user type in the **New User** dialog box.

# Importing the template definition file

After you edit the template definition file, you must re-import it into Security Manager Administration.

When you import the file, Security Manager Administration and Security Manager check the file for errors. If Security Manager Administration finds errors, descriptions of the errors are written to a log file. Any errors found by Security Manager are displayed in a dialog box on-screen.

## To import the template definition file

- 1** Log in to Security Manager Administration. See [“Logging in to Security Manager Administration” on page 46](#).
- 2** Click **File > User Type Templates > Import**.  
The **Open** dialog box appears.
- 3** Select `usertype.templates` in the **Open** dialog box and click **Open**.  
The **Save As** dialog box appears.
- 4** Accept the default or specify a name and location for a log file that lists any errors found by Security Manager Administration when the file is imported, and click **Save**.  
An information dialog box appears to indicate successful file import.
- 5** Click **OK**.

You have now imported the template definition file.

# Testing the template definition file

After you modify the template definition file and re-import it into Security Manager Administration, you should test the file against the **New User** dialog box to ensure that the resulting information is represented correctly.

To guarantee that Entrust PKI administrators can enter users in each user type and can enter the required information in the attribute fields, you should create a test entry in each user type, specifying the required data.

As noted in [“Exporting the template definition file” on page 451](#), you should always keep a backup copy of the `usertype.template` file on disk in a known location. If you experience a problem that you cannot resolve, create a new `usertype.template` file based on the backup and try again.

## To test the template definition file

- 1 Export, edit, and import the `usertype.template` file. See [“Modifying the user template file and user types” on page 447](#).
- 2 Click **Users > New User**.  
The **New User** dialog box appears.

**New User**

Naming | General | Certificate Info | Key Update Options

Type: Person

This is the user type to be used for most Entrust users.

	In DN
* First Name	<input checked="" type="checkbox"/>
* Last Name	<input type="checkbox"/>
Serial Number	<input type="checkbox"/>
Email	<input type="checkbox"/>

Asterisks (\*) appear beside required attributes.

Add to: CA Domain Searchbase

Show DN...

☐ Create profile

OK Cancel Help

- 3** Click the **Type** drop-down list and select a user type.  
Check that the description fields and attribute fields appear correctly.
- 4** Type attribute information in each attribute field for a test entry.  
One attribute field should exist for every attribute added in the templates definition file.
- 5** Click **Show DN** to ensure that at least one attribute is listed.  
All attributes designated in the template definition file for inclusion in the DN should be seen in the DN.
- 6** Click **OK**.
- 7** Repeat [Step 3](#) through [Step 6](#) for each user type listed in the **Type** box.
- 8** Right-click each test entry in the tree view and click Properties in the pop-up menu.  
Review the information for each in the **General Information** property page.

If you are satisfied with the results of the test, you can let Entrust PKI administrators begin adding new users and devices to Security Manager.



## Cross-certifying with other CAs

Cross-certification is a means to establish third-party trust among users who belong to different Certification Authorities (CAs). This third-party trust means that users who belong to one CA can exchange secured data with users in a cross-certified CA. To understand how cross-certification works, you must understand the concepts of third-party trust and direct trust as they relate to network security (see the *Security Manager Deployment Guide*).

This chapter explains how cross-certification relates to trust and how cross-certificates establish trust among CAs. This chapter also describes the types of cross-certification you can perform between Entrust CAs, and how to choose which type of cross-certification you require.

This chapter does not describe how to cross-certify an Entrust CA with a CA from another vendor. Although this type of cross-certification is possible, both CAs must meet special requirements. If you want to perform this type of cross-certification, contact Entrust Customer Support.

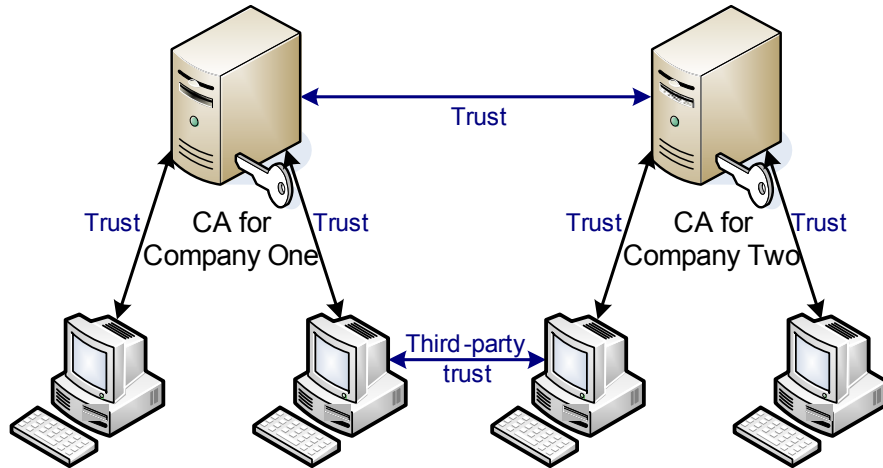
This chapter contains the following sections:

- [“About cross-certification” on page 467](#)
- [“Cross-certifying online” on page 474](#)
- [“Cross-certifying offline” on page 484](#)
- [“Viewing cross-certificates” on page 495](#)
- [“Removing cross-certificates from the directory” on page 497](#)
- [“Publishing cross-certificates to the directory” on page 499](#)
- [“Revoking cross-certificates” on page 501](#)
- [“Taking cross-certificates off hold” on page 504](#)
- [“Updating cross-certificates” on page 505](#)

- “Setting policy constraints requirements in protocol certificates” on page 506

# About cross-certification

Cross-certification is a process that allows two CAs to trust each other. When two CAs trust each other, their users indirectly trust each other. This trust among users from cross-certified CAs is a form of third-party trust. The following diagram shows the trust relationships that exist between cross-certified CAs.



Once trust is established using cross-certification, users can exchange secured information with each other, provided that searchbases are created. A searchbase enables users who belong to a CA to locate users in a cross-certified CA. For more information about searchbases, see [“Administering searchbases” on page 341](#).

This chapter describes how to implement cross-certification. However, there is much more to cross-certification than simply implementation. For example, because cross-certification extends third-party trust, CAs must be completely comfortable with each other's security policies. To use passports as an analogy, one country is unlikely to trust another country's passports without first examining the policies used by that country to create and distribute their passports. Each country would want to understand, in detail, the process by which the other country verifies a citizen's identity before issuing a passport.

As well as understanding the other CA's security policies, a CA should also learn which people have access to high-level security functions for the other CA. It is also likely that representatives of both CAs would sign a legal agreement before performing cross-certification. This agreement would state the required security policies in both domains and give signed assurance that these policies would be followed.

Topics in this section:

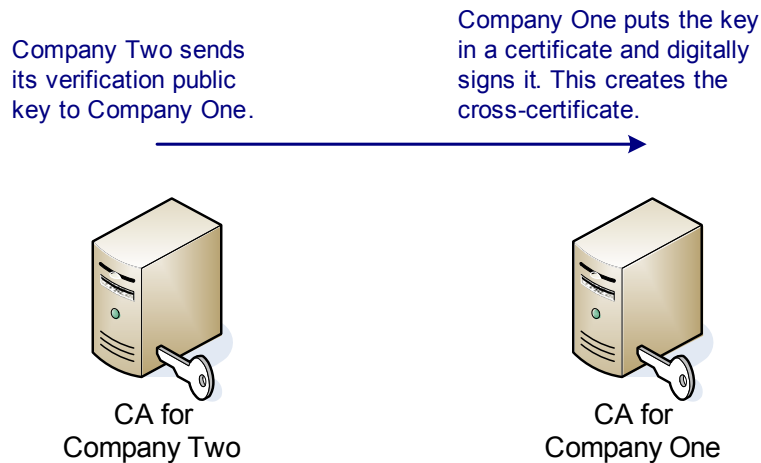
- [“Establishing trust using cross-certification” on page 468](#)
- [“How client applications use cross-certificates” on page 469](#)

- [“Key identifiers in cross-certification” on page 470](#)
- [“Methods for establishing trust” on page 473](#)

## Establishing trust using cross-certification

Within Security Manager, trust is established between CAs using cross-certificates. A cross-certificate contains one CA's verification public key, and is digitally signed by another CA using its own signing private key. The CA that signed the cross-certificate trusts the CA whose verification public key is in the cross-certificate.

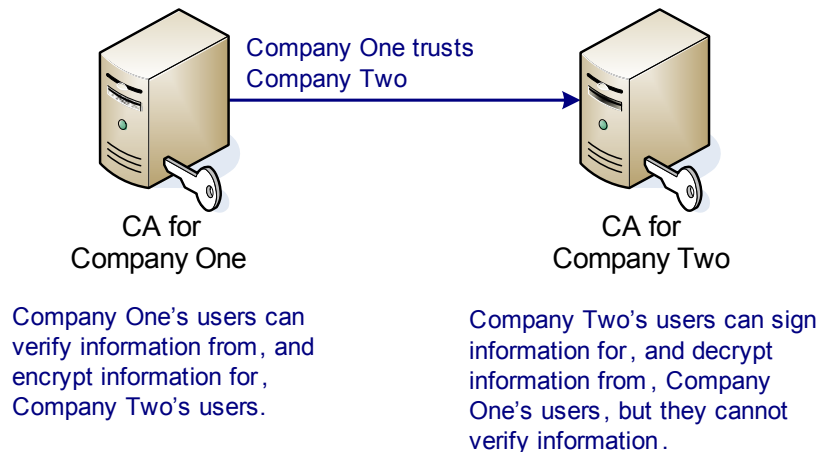
Understanding how trust is established using cross-certificates is best shown by example. Suppose that Company One wants to trust Company Two. As shown in the following diagram, Company Two securely provides its verification public key to Company One. Company One creates a cross-certificate that contains Company Two's key and is digitally signed by Company One. Company One stores the cross-certificate in the CA entry of its directory.



Cross-certificates are valid for 36 months, by default. Security Officers can change the lifetime of cross-certificates using the Administration Policy (see [“Configuring the Security Policy” on page 84](#)). The minimum lifetime is two months and the maximum lifetime is 420 months. You must update cross-certificates before they expire (see [“Updating cross-certificates” on page 505](#)) to ensure there is no interruption in service for users in the cross-certified CAs.

## One-way trust versus two-way trust

Suppose that Company One trusts Company Two, but Company Two does not trust Company One. This relationship is referred to as unilateral cross-certification because trust is established in only one direction (only Company One creates a cross-certificate). As shown in the following diagram, users who belong to Company One indirectly trust users who belong to Company Two, which means that Company One's users can encrypt information for, and verify (authenticate) information from, Company Two's users.



Two-way trust is referred to as mutual cross-certification (that is, both CAs create a cross-certificate). For example, suppose that two departments are managed by their own CAs. Through mutual cross-certification, employees in both departments can encrypt inter-department memos for each other, and verify information from each other.

## How client applications use cross-certificates

When users in cross-certified CAs exchange secured data, Security Manager client applications validate user certificates using cross-certificates. You can see how applications use cross-certificates in the following example, where Bob (of Company One) uses a Security Manager client application to encrypt a file for Alice (of Company Two).

## How Bob encrypts a file for Alice

- 1** Bob selects a file to encrypt for Alice. Bob's client application retrieves Alice's encryption public certificate from the directory.
- 2** The client application checks that Company Two is still trusted using the cross-certificate held by Company One.

The client application verifies the cross-certificate signature, and checks that the cross-certificate has integrity and has not expired. The client application also ensures that the cross-certificate is not revoked by examining the authority revocation list (ARL) signed by Company One.
- 3** The client application extracts Company Two's verification public key from the cross-certificate held by Company One.
- 4** The client application validates Alice's encryption public certificate by validating Company Two's digital signature on the certificate, and checking the certificate's lifetime.

In so doing, the client application checks that the certificate has integrity, and also ensures that the certificate is not revoked by examining the certificate revocation list (CRL) that is issued by Company Two.
- 5** If the cross-certificate held by Company One is valid, and if Alice's encryption public certificate is valid, Bob is allowed to encrypt the file for Alice.

## Key identifiers in cross-certification

There are two key identifier extensions that Security Manager uses when it is searching certificates for the root of trust during client requests:

- the `authorityKeyIdentifier` identifies which key signed a certificate or revocation list

This extension is important for any signed certificate, revocation list, or policy certificate, but is not necessary for self-signed CA root certificates and therefore can be excluded from them. If the `authorityKeyIdentifier` is in a self-signed root certificate, it is in the same format and value as the `subjectKeyIdentifier`.
- the `subjectKeyIdentifier` identifies the public key in a certificate

This extension is included only in certificates, and is important in certificates that contain public keys. It is not necessary for end entity certificates and therefore can be excluded from them.

When a client validates an end entity's certificate, it uses the `authorityKeyIdentifier` in that certificate to find a CA certificate or cross-certificate with a matching `subjectKeyIdentifier`. After the CA certificate is found, the client uses its `authorityKeyIdentifier` and `subjectKeyIdentifier` to find other CA certificates until it gets to its root of trust.

When a CA signs a cross-certificate for another CA, the `authorityKeyIdentifier` in the cross-certificate must exactly match the `subjectKeyIdentifier` in the self-signed root CA certificate that contains the same public key as the cross-certificate. Security Manager does not accept a forward cross-certificate in which the `authorityKeyIdentifier` extension does not match the `subjectKeyIdentifier` in the associated root CA certificate.

## Changing the format of key identifiers

When validating certificates, two key identifier extensions are used: `subjectKeyIdentifier` and `authorityKeyIdentifier`. The `subjectKeyIdentifier` is a hash of the public key in a certificate and identifies the public key. The `authorityKeyIdentifier` is a hash of the public key that matches the private key that signed the certificate and identifies the signing key.

For example, consider the following self-signed root CA certificates:

- self-signed root certificate for CA1

```
{
    public key A
    subjectKeyIdentifier = identifierA
    authorityKeyIdentifier = identifierA
} signed by private key A
```
- self-signed root certificate for CA2

```
{
    public Key B
    subjectKeyIdentifier = identifierB
    authorityKeyIdentifier = identifierB
} signed by private key B
```

If CA1 certifies CA2, the cross-certificate issued by CA1 is:

```
{
    public Key B
    subjectKeyIdentifier = identifierB
    authorityKeyIdentifier = identifierA
} signed by private key A
```

If CA2 certifies CA1, the cross-certificate issued by CA2 is:

```
{
    public Key A
    subjectKeyIdentifier = identifierA
    authorityKeyIdentifier = identifierB
```

```
} signed by private key B
```

In some cases, the CA creating a cross-certificate does not receive a copy of the `subjectKeyIdentifier` extension from the other CA and calculates the `subjectKeyIdentifier` using the other CA's public key. When this occurs, the CA creating the cross-certificate must use the same method to create the `subjectKeyIdentifier` as used in the root CA certificate that contains the public key included in the cross-certificate.

Change how the Certification Authorities (CAs) format key identifiers as follows:

- If you are cross-certifying with Security Manager 7.x or later, you do not have to change the key identifier format. Security Manager 7.x and later include the `subjectKeyIdentifier` extension in the cross-certification request.
- If you are cross-certifying with Security Manager 6.x or earlier, each CA must format its key identifier to be the same as the format used in the latest CA certificate at the other CA.

This is necessary because Security Manager 6.x or earlier does not include or calculate the `subjectKeyIdentifier` extension in cross-certification requests and therefore each CA must calculate the `subjectKeyIdentifier` extension from the public key in the cross-certification request. The `subjectKeyIdentifier` in a cross-certificate identifies the public key in the certificate and this must match the `authorityKeyIdentifier` in the CA certificate which holds the public key.

You change how Security Manager formats key identifiers by setting the `KeyIdMode` variable in Security Manager Control Command Shell. By default, `KeyIdMode=0` in Security Manager 6.x and earlier, and `KeyIdMode=1` in Security Manager 7.x or later. When changing the `KeyIdMode` variable, you must change `KeyIdMode` on each Security Manager to match the other setting used in the other Security Manager's latest CA certificate.

For example, if you are cross-certifying with a Security Manager 6.0 CA and the default `KeyIdMode` has not been changed at either CA, you must change the Security Manager 6.0 `KeyIdMode` to 1 and your Security Manager `KeyIdMode` to 0. After cross-certification is complete, change both `KeyIdMode` settings back to the original value.

---

**Note:** Changing `KeyIdMode` at Security Manager does not change the format of the key identifiers in the CA certificate until the next CA key update. If you previously changed `KeyIdMode` and are unsure of the key identifier format in your CA certificate, contact Entrust customer support.

---

For details about viewing and changing the `KeyIdMode` variable, see the *Security Manager Operations Guide*.



- If you are creating a cross-certificate for a third-party CA, and the third-party CA does not include the `subjectKeyIdentifier` in cross-certification requests, you have these options:
  - Determine the key identifier format used in the third-party root CA certificate and set the `KeyIdMode` at Security Manager to match it.  
If the third-party vendor cannot indicate the format used based on the description of the `KeyIdMode` format, contact Entrust Support. After cross-certification is complete, change the `KeyIdMode` setting back to its original value.
  - Create a cross-certificate for the third-party CA using the third-party CA root CA certificate instead of a cross-certificate request. Use the `-cert` option of the `ca xcert create` command in the Security Manager Control Command Shell (see the *Security Manager Operations Guide* for details).
- If a third-party CA is creating a cross-certificate for your Security Manager and the third-party CA does not use the `subjectKeyIdentifier` that Security Manager includes in the cross-certification request, contact your third-party CA vendor and ask what options are available to ensure that the `subjectKeyIdentifier` is calculated properly.

A third-party CA must support one of the following to create a cross-certificate with the correct `subjectKeyIdentifier` extension:

- The third-party CA can copy the `subjectKeyIdentifier` extension from the PKCS#10 cross-certificate request and include it in the cross-certificate.
- The third-party CA can calculate the `subjectKeyIdentifier` extension in the same format as in the Security Manager root CA certificate.
- The third-party CA can create a cross-certificate using a root CA certificate and copies the `subjectKeyIdentifier` extension from the root CA certificate into the cross-certificate.

## Methods for establishing trust

You can establish trust between CAs online or offline.

- Online cross-certification occurs between CAs on servers that connect over a network through TCP/IP. For more information about cross-certifying CAs online, see [“Cross-certifying online” on page 474](#).
- Offline cross-certification occurs between CAs on servers that do not connect over a network using TCP/IP (for example, a network is not available or you do not want the servers connected over a network). For more information about cross-certifying CAs offline, see [“Cross-certifying offline” on page 484](#).

For troubleshooting information, see the *Security Manager Operations Guide*.

# Cross-certifying online

Online cross-certification is a method of cross-certifying CAs on servers that are connected over a network. You can perform unilateral cross-certification online to establish one-way trust and you can perform mutual cross-certification online to establish two-way trust.

The information in this section assumes that the necessary trusting business relationships (such as signed legal agreements) are already established.

---

**Attention:** You cannot cross-certify with a Security Manager 6.x or earlier CA if either CA uses an elliptic curve. Security Manager cannot read hex-encoded CRLs from an HTTP location; Security Manager can read only binary-encoded CRLs from an HTTP location.

---

Topics in this section:

- [“Online cross-certification requirements” on page 474](#)
- [“Communication between CAs” on page 475](#)
- [“Performing online cross-certification” on page 475](#)

## Online cross-certification requirements

Before you begin online cross-certification

- the CAs must be located on servers that are connected over a network.  
The servers must be connected through TCP/IP during the cross-certification process. Once cross-certification is complete, the network connection is no longer necessary.
- the directories for both CAs must be connected.  
The directories can be connected physically (for example, they could be two branches in one directory) or logically (for example, they could be two physically distinct directories that are chained together). See [“Preparing the directory” on page 523](#).

After you complete online cross-certification

- the directories must remain connected to enable Entrust applications to verify the validity of user certificates in the directory.
- the CAs must have searchbases to enable Entrust applications to locate users who belong to a cross-certified CA. For more information about searchbases, see [“Administering searchbases” on page 341](#).

# Communication between CAs

During online cross-certification, the two CAs communicate with each other. For example, the CAs communicate when one CA provides its verification public key to the other CA for inclusion in a cross-certificate. The two CAs must communicate using PKIX-CMP (Public-Key Infrastructure X.509–Certificate Management Protocol), an industry-standard protocol defined by the IETF (Internet Engineering Task Force). For information about PKIX-CMP, contact the IETF and request RFC 2510. PKIX-CMP supports mutual and unilateral cross-certification.

## Performing online cross-certification

To perform online unilateral or mutual cross-certification, one CA follows a procedure to initiate cross-certification and the other CA follows a procedure to complete cross-certification.

For mutual cross-certification, either CA can initiate cross-certification because trust is established in both directions. However, because unilateral cross-certification establishes trust only one way, the CA that wants to trust must initiate the unilateral cross-certification, and the CA that is to be trusted must complete it.

The procedures in this chapter refer to the company that wants to trust as Company One, and the company that wants to be trusted as Company Two.

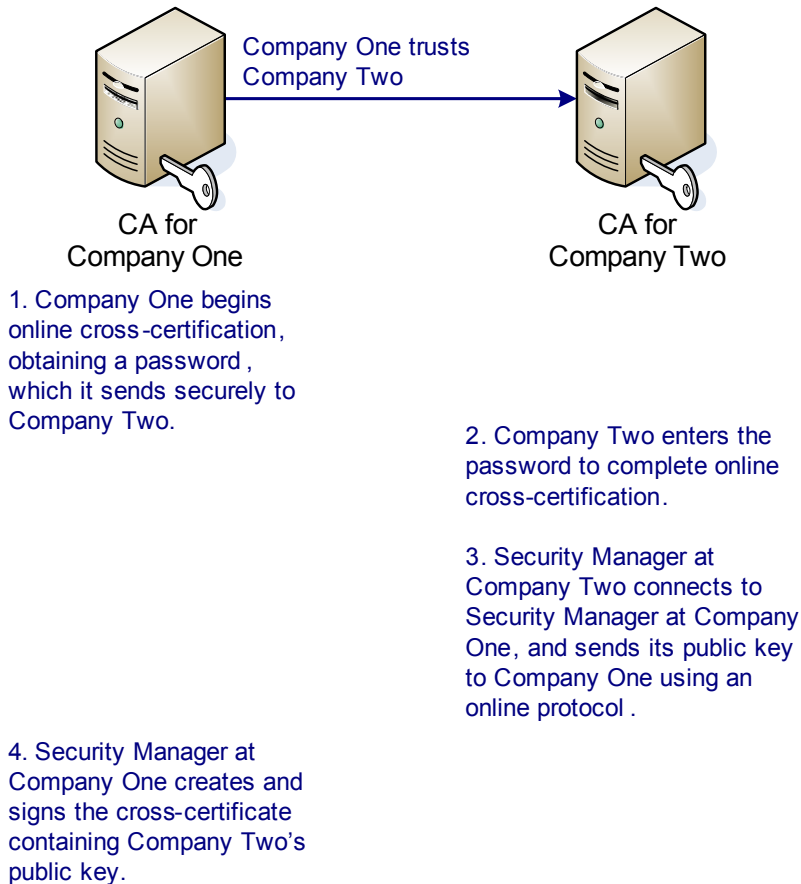
**Note:** If you perform mutual (two-way) cross-certification, you will perform the procedures twice. The second time, Company Two will be the Company that wants to trust and Company One will be the company that wants to be trusted.

Table 45 on page 475 shows the online cross-certification characteristics of each company.

**Table 45:** Online cross-certification characteristics

Company One...	Company Two...
<ul style="list-style-type: none"><li>wants to trust Company Two</li><li>begins the unilateral cross-certification process</li><li>signs a cross-certificate containing Company Two's public key</li><li>certifies Company Two</li></ul>	<ul style="list-style-type: none"><li>wants to be trusted by Company One</li><li>completes the unilateral cross-certification process</li><li>sends its verification public key to Company One</li><li>is certified by Company One</li></ul>

There are four main steps in the unilateral online cross-certification process:



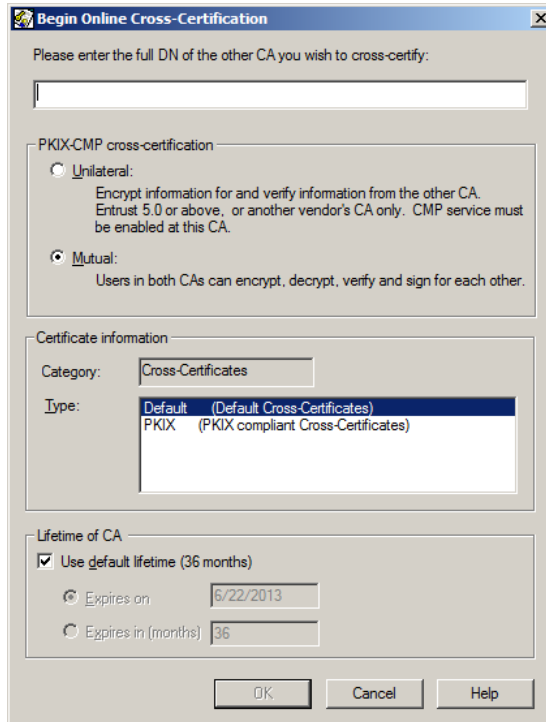
The following procedures describe how to

- initiate cross-certification
- obtain the password that is used to complete cross-certification
- complete cross-certification

#### To initiate online cross-certification at Company One

- 1** Log in to Security Manager Administration for Company One (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Select **CAs > Cross-certified CAs > Online Cross-Certification > Begin Online Cross-Certification**.

The **Begin Online Cross-Certification** dialog box appears.

The image shows a Windows-style dialog box titled "Begin Online Cross-Certification". It contains several sections: a text input field for the "full DN of the other CA"; a "PKIX-CMP cross-certification" section with radio buttons for "Unilateral" and "Mutual"; a "Certificate information" section with a "Category" dropdown set to "Cross-Certificates" and a "Type" list box showing "Default (Default Cross-Certificates)" and "PKIX (PKIX compliant Cross-Certificates)"; and a "Lifetime of CA" section with a checked "Use default lifetime (36 months)" option and fields for "Expires on" (6/22/2013) and "Expires in (months)" (36). At the bottom are "OK", "Cancel", and "Help" buttons.

- 3** In the **Begin Online Cross-Certification** dialog box:
  - a** In the text box at the top of the dialog box, enter the DN of the CA you are cross-certifying with (for example, `dc=Company Two,dc=com`).
  - b** Under **PKIX-CMP cross-certification**, select the type of cross-certification you want to perform (**Unilateral** or **Mutual**).
  - c** Under **Certificate information**, select the type of cross-certificate.  
Only the existing cross-certificate types appear. For information about cross-certificate types, see ["Customizing CA certificates" on page 591](#).
  - d** Under **Lifetime of CA**, set the lifetime of the cross-certificate:
    - To use the default lifetime from the security policy select **Use default lifetime**. By default this option is selected.
    - To specify an expiry date, deselect **Use default lifetime**, then click **Expires on** and enter the expiry date in the form `MM/DD/YYYY`.
    - To specify a lifetime (in months), deselect **Use default lifetime**, then click **Expires in (months)** and enter the lifetime, from 2 to 420 months.
  - e** Click **OK**.

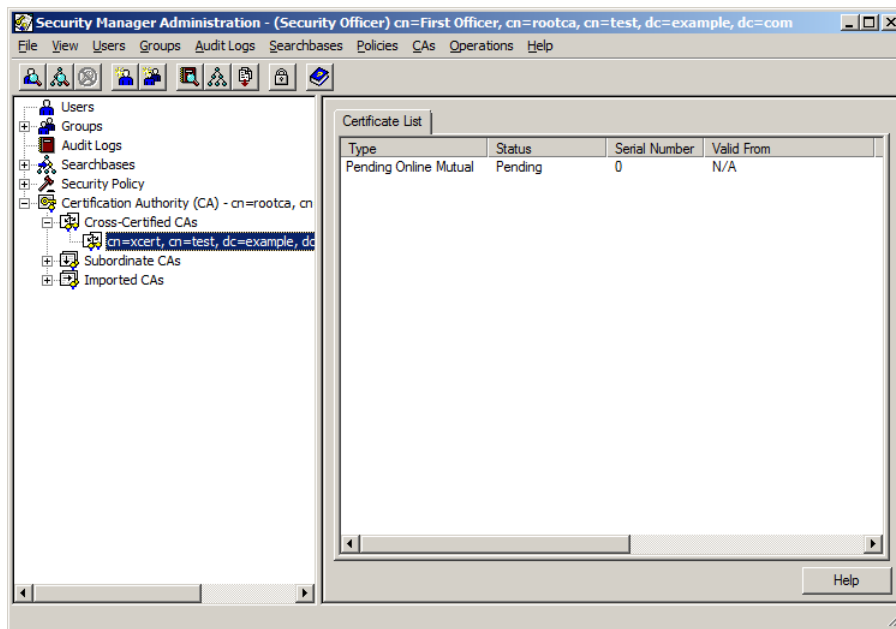
- 4 If prompted, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

A dialog box appears when the initiation is complete. Record the password displayed on this dialog box. You need to give this password to the administrator of the other CA (Company Two).

- 5 Click **OK**.

You have now initiated online cross-certification. If you did not obtain the password required to complete online cross-certification, see [“To obtain the cross-certification password” on page 479](#) for details about obtaining the password.

The CA that completes cross-certification (that is, Company Two) appears in Company One's Security Manager Administration window with a status of **Pending**.



---

**Note:** When Company Two completes cross-certification, the status in Company One's Security Manager Administration window will change from **Pending** to **Complete**. You must select CAs > Refresh to see the change.

---

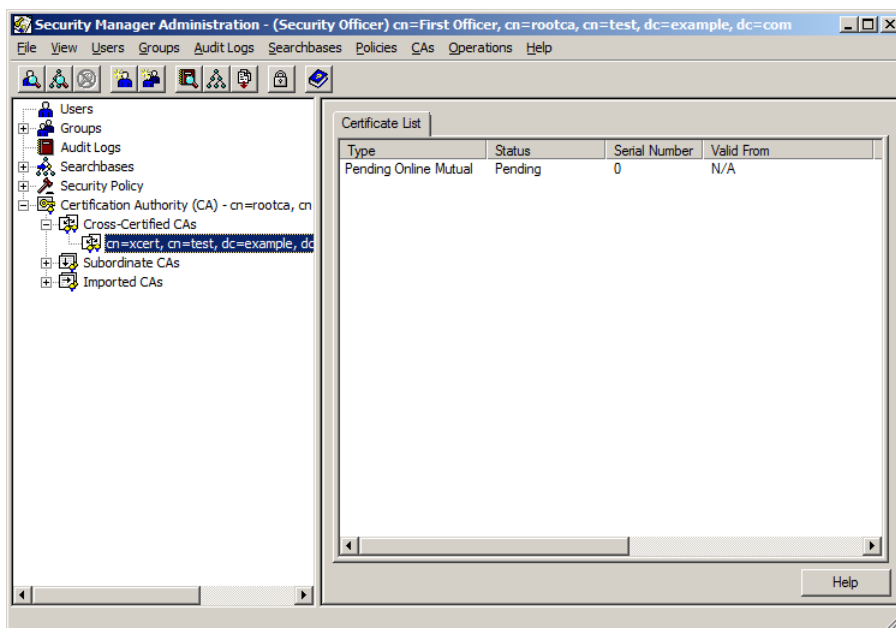
You must provide the Security Officer for the CA that completes cross-certification with the:

- TCP/IP address and port number of the CA that initiated cross-certification (Company One)
- password used to secure the certification process

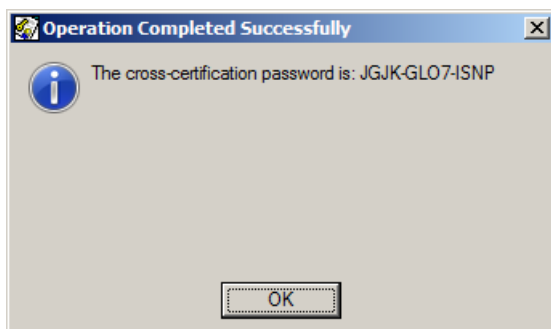
You must provide this password to the Security Officer in a highly secure manner, such as in person or by secure email.

### To obtain the cross-certification password

- 1 Log in to Security Manager Administration for Company One (see [“Logging in to Security Manager Administration”](#) on page 46).
- 2 In the tree view, expand **Certification Authority > Cross-Certified CAs** and then select the DN of the other CA.



- 3 In the **Certificate List** property page, double-click **Pending Online Mutual**. An **Operation Completed Successfully** dialog box appears, showing the 12-character single-use password required for cross-certification.



You have now obtained the cross-certification password.

---

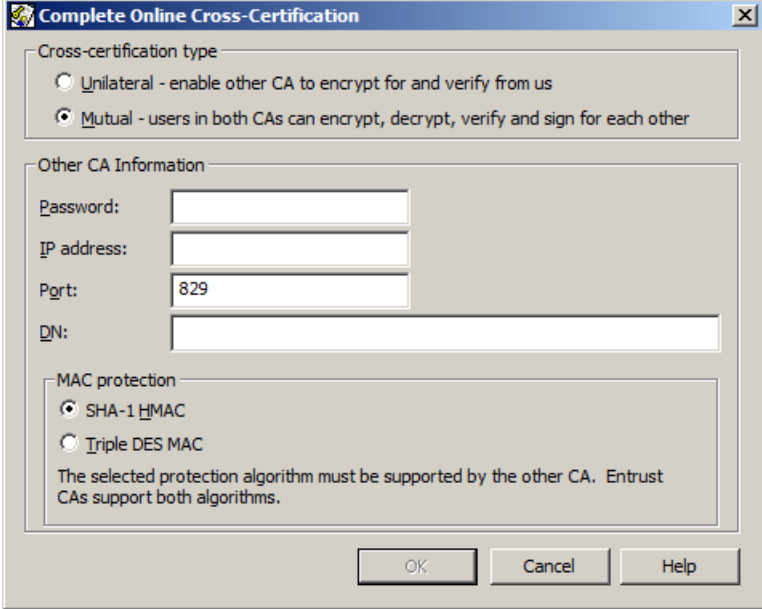
**Note:** This password is valid for three days only. If the other CA does not complete cross-certification within three days, you must cancel the cross-certification and then repeat the initiation procedure.

---

### To complete cross-certification online at Company Two

- 1 Ensure that you have received the following information from the Security Officer for the CA that initiated cross-certification:
  - TCP/IP address and port number of the CA that initiated cross-certification (that is, Company One)
  - password used to secure the certification process
- 2 Log in to Security Manager Administration for Company Two (see [“Logging in to Security Manager Administration” on page 46](#)).
- 3 Select **CAs > Cross-Certified CAs > Online Cross-Certification > Complete Online Cross-Certification**.

The **Complete Online Cross-Certification** dialog box appears.



The dialog box titled "Complete Online Cross-Certification" contains the following sections:

- Cross-certification type:** Two radio buttons are present. The first is "Unilateral - enable other CA to encrypt for and verify from us". The second, "Mutual - users in both CAs can encrypt, decrypt, verify and sign for each other", is selected.
- Other CA Information:** Four text input fields are provided for "Password:", "IP address:", "Port:" (with the value "829" entered), and "DN:".
- MAC protection:** Two radio buttons are present. The first, "SHA-1 HMAC", is selected. The second is "Triple DES MAC". Below these buttons, a note states: "The selected protection algorithm must be supported by the other CA. Entrust CAs support both algorithms."

At the bottom of the dialog box are three buttons: "OK", "Cancel", and "Help".

- 4 In the **Complete Online Cross-Certification** dialog box, do the following:
  - a Under **Cross-certification type**, select the type of cross-certification.



- b** In the **Password** field, type the cross-certification password that you received from the Security Officer for Company One (see [“To obtain the cross-certification password”](#) on page 479).

When you type the password, the hyphens (-) are optional. The hyphens are present in the password solely for readability.

- c** In the **IP address** field, enter the fully qualified host name or IP address of the server hosting Company One.
- d** In the **Port** field, type the port number of Company One.
- e** In the **DN** field, type the DN for Company One.
- f** Under **MAC protection**, choose **SHA-1 HMAC** if you are cross-certifying with a CA that uses SHA-1 HMAC protection, or **Triple DES MAC** if you are cross-certifying with a CA that uses Triple DES MAC protection.

These message authentication codes are used when authenticating between two CAs that are cross-certifying using PKIX-CMP.

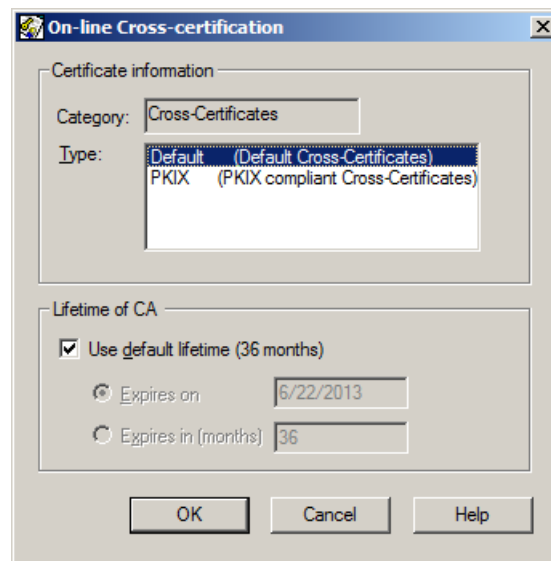
---

**Note:** If you are performing mutual cross-certification with a pre-8.1 version of Security Manager, you must select **Triple DES MAC** or the cross-certification will fail.

---

- g** Click **OK**.

The **On-line Cross-certification** dialog box appears.



- 5** In the **On-line Cross-certification** dialog box:
- a** Click the type of certificate that you want to create.

Only the existing cross-certificate types appear. For information about cross-certificate types, see [“Customizing CA certificates” on page 591](#).

- b** Set the lifetime of the cross-certificate:
  - To use the default lifetime configured in the security policy, select **Use default lifetime**. By default, this option is already selected.
  - To set an expiry date, deselect **Use default lifetime**, then click **Expires on** and then enter the expiry date in the form MM/DD/YYYY.
  - To enter a lifetime (in months) for the cross-certificate, deselect **Use default lifetime**, then click **Expires in (months)** and then enter the lifetime (from 2 to 420 months) for the cross-certificate.
- c** Click **OK**.

- 6** If prompted, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

A dialog box appears when the cross-certification is complete.

- 7** Click **OK**.

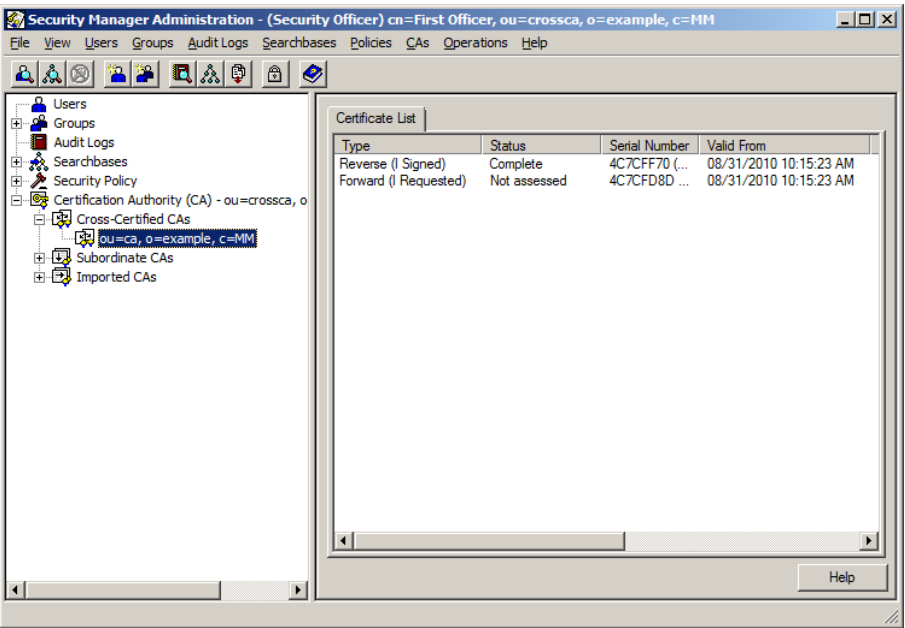
You have now completed the cross-certification process. Security Manager sends its public verification key over the communication link to the other CA at Company One. The Company One CA responds by creating and signing the cross-certificate containing your CA's public key.

---

**Note:** Cross-certificates are valid for 36 months, by default (see [“Configuring the Security Policy” on page 84](#)). You must update cross-certificates before they expire (see [“Updating cross-certificates” on page 505](#)) to ensure there is no interruption in service for users in the cross-certified CAs.

---

If Company One and Company Two performed mutual cross-certification, each cross-certificate appears in the other CA's Security Manager Administration window with a status of **Complete**. For example:



If Company One and Company Two have performed unilateral cross-certification, Company Two appears in the Security Manager Administration window for Company One with a status of Complete. (Company One does not appear in the Security Manager Administration window for Company Two because trust is established only one way).

You can click a cross-certified CA in the Security Manager Administration window to display information about that CA. For details on displaying information about a cross-certified CA, see [“Viewing cross-certificates” on page 495](#).

# Cross-certifying offline

Offline cross-certification is a method of cross-certifying CAs on servers that are not connected over a network through TCP/IP. You can perform unilateral cross-certification offline to establish one-way trust or you can perform mutual cross-certification offline to establish two-way trust.

The information in this section assumes that the necessary trusting business relationships (such as signed legal agreements) are established.

---

**Attention:** You cannot cross-certify with a Security Manager 6.x or earlier CA if either CA uses an elliptic curve. Security Manager cannot read hex-encoded CRLs from an HTTP location; Security Manager can read only binary-encoded CRLs from an HTTP location.

---

Topics in this section:

- [“Offline cross-certification requirements” on page 484](#)
- [“Performing offline cross-certification” on page 484](#)

## Offline cross-certification requirements

Before you begin offline cross-certification, connect the directories for both CAs. You can connect the directories physically (for example, two branches in one directory) or logically (for example, two physically distinct directories that are chained together).

After you complete offline cross-certification,

- the directories must remain connected to enable Entrust desktop applications to verify the validity of user certificates in the directories.
- the CAs must have searchbases to enable Entrust desktop applications to locate users who belong to a cross-certified CA. For more information about searchbases, see [“Administering searchbases” on page 341](#).

---

**Attention:** When performing an offline cross-certification, a subordinate CA looks for the root CA's entry. If the subordinate CA cannot find or access the root CA's entry, an Object Not Found error results. To avoid this problem, ensure that the directories to which each CA points are chained or referred. For more information, see [“Chaining and referring directories” on page 523](#).

---

## Performing offline cross-certification

- The procedures in this section show you how to perform unilateral and mutual cross-certification:

- To perform unilateral (one-way) offline cross-certification, one CA follows a procedure to initiate cross-certification and the other CA follows a procedure to complete cross-certification. Because unilateral cross-certification establishes trust only one way, it is important that the CA that initiates cross-certification is the one that is trusted by the CA that completes cross-certification.
- To perform mutual (two-way) offline cross-certification, the initiation and completion procedures are performed twice—once to establish trust in one direction (for example, from Company One to Company Two) and once to establish trust in the reverse direction (for example, from Company Two to Company One). Because you plan to establish trust in both directions, either CA can initiate cross-certification first.

After you complete cross-certification, the trusted CA must import the signed cross-certificate from the trusting CA into its database, and push it into its directory.

The procedures in this chapter refer to the company that wants to trust as Company One, and the company that wants to be trusted as Company Two.

---

**Note:** If you perform mutual cross-certification, you will perform the procedures twice. The second time, Company Two will be the Company that wants to trust and Company One will be the company that wants to be trusted.

---

There are five steps in the unilateral offline cross-certification process:



1. Company Two begins offline cross-certification by creating a PKCS#10 request that contains its verification public key.

2. Company Two sends the certificate request and a validation string securely to Company One.

3. Company One verifies the validation string, and signs the cross-certificate containing Company Two's public key.

4. Company One exports the certificate, and delivers it securely to Company Two.

5. Company Two imports the cross-certificate into its database.

The following procedures describe how to:

- initiate cross-certification
- complete cross-certification
- export the cross-certificate
- import the cross-certificate

---

**Note:** Once you complete these procedures, the cross-certificate is automatically added to the directory of Company One.

---

## To initiate offline cross-certification at Company Two

- 1 Log in to Security Manager Administration for Company Two (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Select **CAs > Cross-Certified CAs > Offline Cross-Certification > Request Cross-Certificate for > Enterprise/Web**.

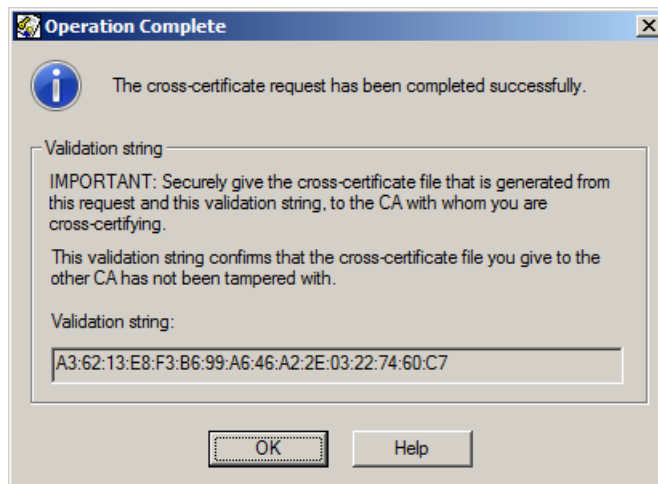
The **CA Cross-Certificate Request** dialog box appears.

- 3 Choose a destination folder and type a name for your cross-certificate request file.

You can choose the file format to be binary (an extension of `.der` is appended to the file name), or PEM (an extension of `.pem` is appended to the file name). The choice of formats is provided for interoperability with PKIs from other vendors. Security Manager can read files in either format.

- 4 Click **Save**.
- 5 If prompted, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

When the initiation is complete, a dialog box appears displaying the validation string. The Security Officer (or an Entrust PKI administrator with appropriate permissions) at Company One verifies the validation string to confirm that the cross-certificate request file has not been tampered with.



- 6 Record the validation string and click **OK**.

This validation string is created based on an MD5 hash of the PKCS #10 request.

---

**Note:** This validation string is calculated on the entire PKCS#10 request. For a validation string that is calculated only on the root CA certificate within the request, use the Security Manager Control Command Shell. See the *Security Manager Operations Guide*.

---

- 7** Deliver the cross-certificate request (that is, the file with the .der or .pem extension) file to Company One in a secure manner.

You have now initiated offline cross-certification.

### To complete offline cross-certification at Company One

- 1** Log in to Security Manager Administration for Company One (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Select **CAs > Cross-Certified CAs > Offline Cross-Certification > Sign Cross-Certificate for > Enterprise/Web**.

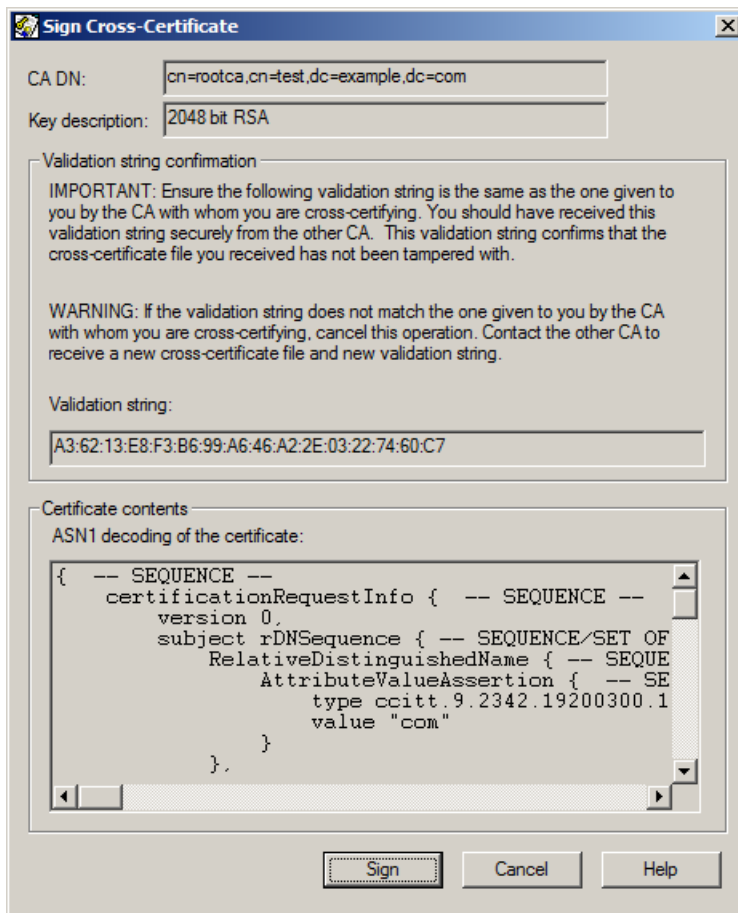
The **CA Cross-Certificate Request** dialog box appears.

- 3** Locate and click the cross-certificate request file and click **Open**.

The cross-certificate request file was provided to you by a Security Officer of the CA you are cross-certifying with. The file has either a .der or .pem extension (for example, company2crosscert.der).

The **Sign Cross-Certificate** dialog box appears.





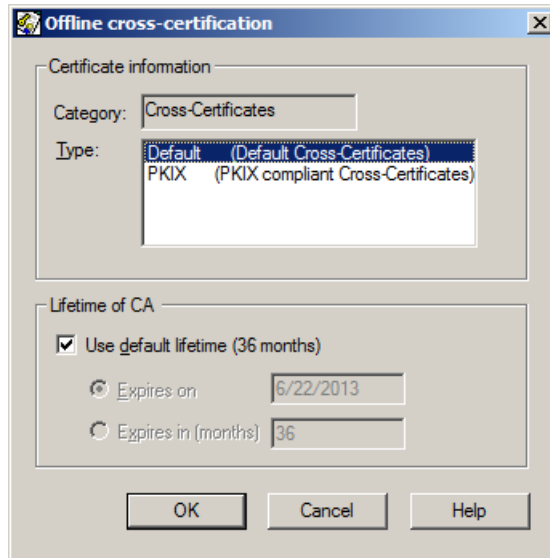
- 4 Compare the validation string in the dialog box to the validation string that a Security Officer of the other CA gave you.

This validation string is created based on an MD5 hash of the PKCS #7 response.

If the validation strings do not match, click **Cancel**. The cross-certificate request file may have been tampered with. Ask the Security Officer of the other CA for a new cross-certificate request file.

- 5 If the validation strings match, click **Sign**.

The **Offline cross-certification** dialog box appears.



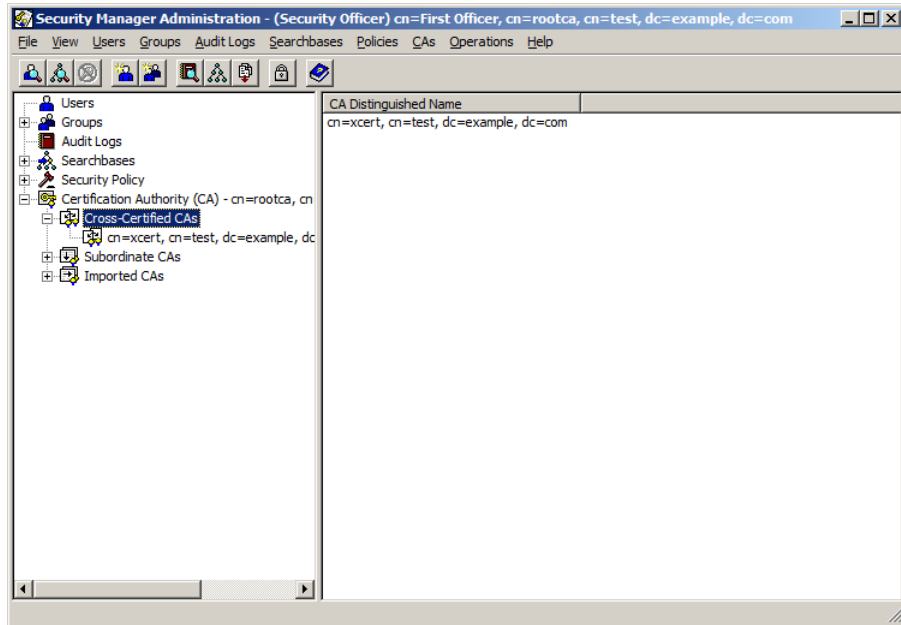
- 6 In the **Offline cross-certification** dialog box, do the following:
  - a Click the type of certificate that you want to create.  
Only the existing cross-certificate types appear. For information about cross-certificate types, see [“Customizing cross-certificates” on page 580](#).
  - b Set the lifetime of the cross-certificate:
    - To use the default lifetime configured in the security policy, select **Use default lifetime**. By default, this option is already selected.
    - To set an expiry date, deselect **Use default lifetime**, then click **Expires on** and then enter the expiry date in the form MM/DD/YYYY.
    - To enter a lifetime (in months) for the cross-certificate, deselect **Use default lifetime**, then click **Expires in (months)** and then enter the lifetime (from 2 to 420 months) for the cross-certificate.
  - c Click **OK**.
- 7 If prompted, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
An **Operation Completed Successfully** dialog box appears when the cross-certification is complete.
- 8 Click **OK**.  
You have now signed a cross-certificate.

---

**Note:** Cross-certificates are valid for 36 months, by default (see [“Configuring the Security Policy” on page 84](#)). You must update cross-certificates before they expire to ensure there is no interruption in service for users in the cross-certified CAs (see [“Updating cross-certificates” on page 505](#)).

---

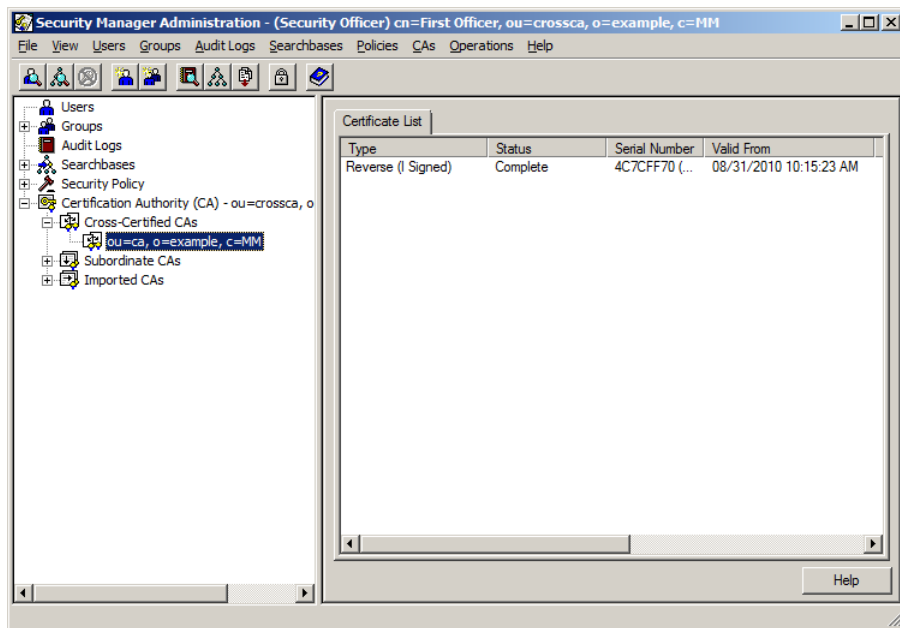
The CA that initiated cross-certification appears in the Security Manager Administration window under **Cross-Certified CAs**.



### To export the cross-certificate to a file

- 1 Log in to Security Manager Administration for Company One (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 In the tree view, expand **Certification Authority** > **Cross-Certified CAs** and then select the DN of the other CA.

A list of cross-certificates for that CA appear. Cross-certificates you signed for other CAs are called reverse cross-certificates.



- 3 Right-click the cross-certificate you want to export and select **Write this cross-certificate to file**.

The **Write Cross Certificate to File** dialog box appears.

- 4 Choose a destination folder and type a name for the cross-certificate file.

You can choose to export the certificate as raw certificate data, or you can export the certificate in the PKCS#7 format, as defined by the PKCS#7 standard. You can also choose the file format to be binary (an extension of `.der` is appended to the filename), or PEM (an extension of `.pem` is appended to the filename). The export options and the file formats are provided for interoperability with PKIs from other vendors. Security Manager can read certificates exported in any combination of the available export options and file formats.

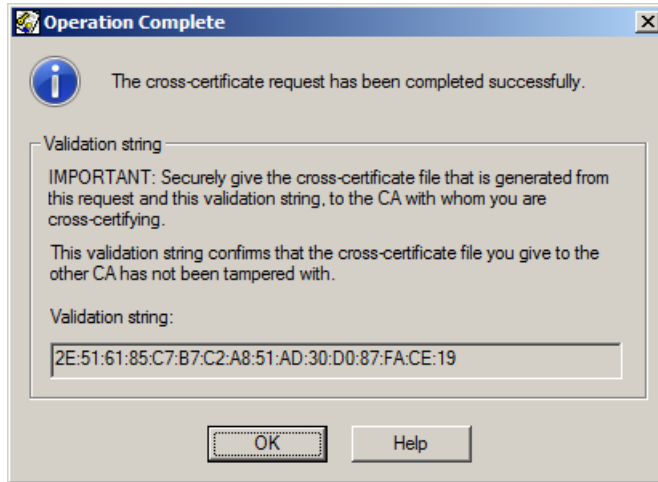
---

**Note:** If you are exporting a forward cross-certificate, Security Manager does not have the ability to export it using the PKCS#7 format, therefore it is exported as raw certificate data. In this case, the **Export options** on the **Write Cross-Certificate to File** dialog box is unavailable.

---

- 5 Click **Save**.

When the cross-certificate is written to file, a dialog box appears displaying the validation string. The Security Officer at Company Two verifies the validation string to confirm that the cross-certificate file has not been tampered with.



---

**Note:** This validation string is calculated on the entire PKCS#7 request. For a validation string that is calculated only on the root CA certificate within the file, use the Security Manager Control Command Shell. See the *Security Manager Operations Guide*.

---

- 6** Record the validation string on paper and click **OK**.
- 7** Deliver the cross-certificate file (the DER or PEM file) to Company Two in a secure manner.

You have now exported the cross-certificate signed by Company One.

#### To import the cross-certificate

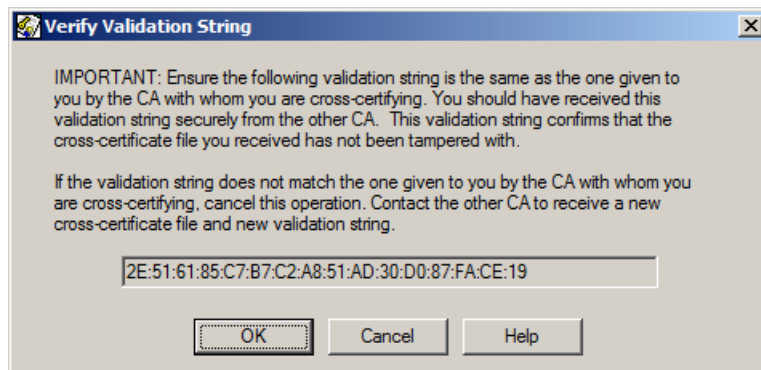
- 1** Log in to Security Manager Administration for Company Two (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** In the tree view, expand **Certification Authority**.
- 3** Right-click **Cross-Certified CAs**, and then select **Offline Cross-Certification > Import cross-certificate I requested**.

The **Import Cross-Certificate I Requested** dialog box appears.

- 4** Locate and click the cross-certificate file from Company One and click **Open**.

The cross-certificate file was provided to you by a Security Officer of Company One. The file has a `.der` or a `.pem` extension (for example, `CA2signed_byCA1.der`).

The **Verify Validation String** dialog box appears.



- 5 Compare the validation string in the dialog box to the validation string that a Security Officer of the other CA gave you.  
If the validation strings do not match, click **Cancel**. The cross-certificate file may have been tampered with. Ask the Security Officer of the other CA for a new cross-certificate file.
- 6 If the validation strings match, click **OK**.
- 7 If prompted, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

When the import is complete, a dialog box appears confirming successful completion of the operation.

---

**Note:** Once you complete this procedure, the cross-certificate is automatically added to the directory of Company Two.

---

- 8 Click **OK**.

You have now imported the cross-certificate signed by Company One.

---

**Note:** Cross-certificates are valid for 36 months, by default (see [“Configuring the Security Policy” on page 84](#)). You must update cross-certificates before they expire (see [“Updating cross-certificates” on page 505](#)) to ensure there is no interruption in service for users in the cross-certified CAs.

---

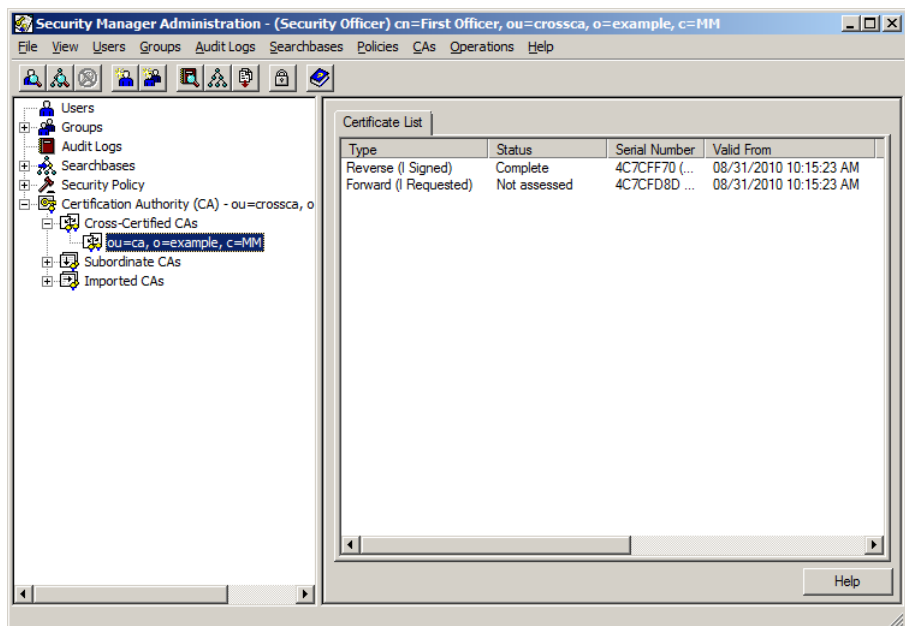
# Viewing cross-certificates

You can use Security Manager Administration to view cross-certificates. Two types of cross-certificates can be related to a CA: forward and reverse cross-certificates. A forward cross-certificate is a cross-certificate created and signed by another CA, which contains your CA's verification public key. A reverse cross-certificate is a cross-certificate that your CA created and signed, which contains another CA's verification public key.

## To view cross-certificates

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration”](#) on page 46).
- 2 In the tree view, expand **Certification Authority > Cross-Certified CAs** and then select the DN of the other CA.

A list of forward and reverse cross-certificates for that CA appear in the Security Manager Administration window.



The property page displays the following information:

- **Type** specifies the type of cross-certificate:
  - For forward cross-certificates (cross-certificates signed by another CA), the type is **Forward (I requested)**.
  - For reverse cross-certificates (cross-certificates signed by your CA), the type is **Reverse (I signed)**.

- If you initiated mutual or unilateral cross-certification online, and the other CA has not completed the cross-certification, the type is either **Pending Online Mutual** or **Pending Online Unilateral**.
  - **Status** contains the state of the cross-certificate:
    - The state **Complete** indicates that the reverse cross-certificate is signed or received successfully.
    - The state **Revoked** means the cross-certificate is revoked and is no longer valid (see [“Revoking cross-certificates” on page 501](#)).
    - The state of forward cross-certificates are **Not assessed**, because forward cross-certificates are unmanaged certificates.
    - If you initiated mutual or unilateral cross-certification online, and the other CA has not completed the cross-certification, the status is **Pending**.
  - **Serial number** contains the serial number and unique ID of the cross-certificate.
  - **Valid from** contains the date and time that the cross-certificate was signed.
  - **Expires** contains the date and time that the cross-certificate expires. See [“Updating cross-certificates” on page 505](#) for information on how to update a cross-certificate that is about to expire.
  - **Published in Directory** specifies whether the cross-certificate is published in the Security Manager directory.

When Security Manager imports a forward cross-certificate or signs a reverse cross-certificate, Security Manager automatically saves the cross-certificate in the Security Manager database and in the directory. You can remove cross-certificates from the directory (see [“Removing cross-certificates from the directory” on page 497](#)).
- 3** If you want detailed information about a specific cross-certificate, double-click the cross-certificate. For an explanation of the certificate information, see [“Configuring user properties” on page 219](#).



# Removing cross-certificates from the directory

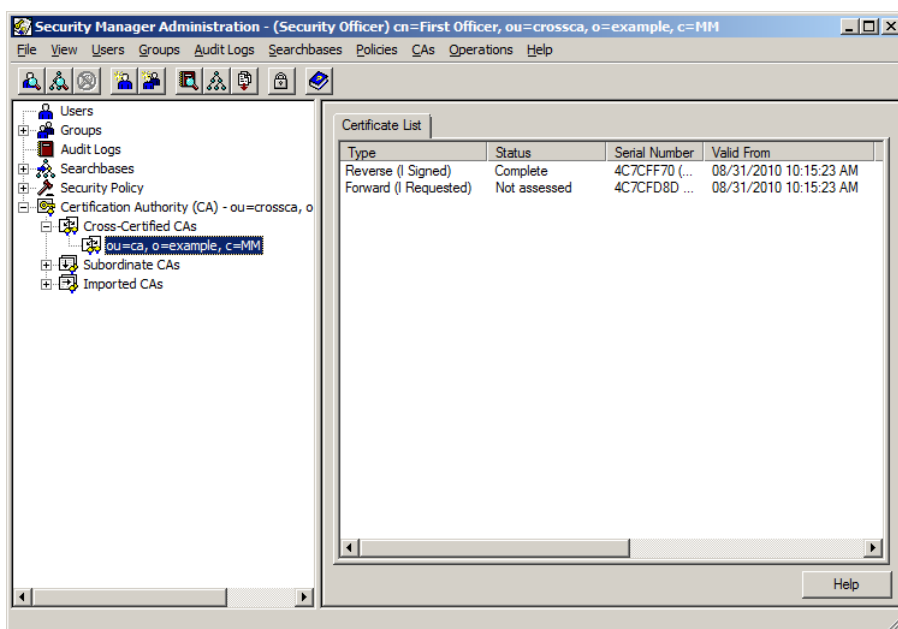
When you import a forward cross-certificate, Security Manager automatically saves the cross-certificate in the Security Manager database and in the directory. When your CA issues a reverse cross-certificate, Security Manager automatically saves the cross-certificate in the database and directory.

If required, you can remove cross-certificates from the directory. For example, you may want to remove a forward cross-certificate from the directory if the other CA revokes its cross-certificate.

## To remove a cross-certificate from the directory

- 1 Log in to Security Manager Administration as a Security Officer or an Entrust PKI administrator with appropriate permissions.
- 2 In the tree view, expand **Certification Authority > Cross-Certified CAs** and then select DN of the other CA.

The cross-certified CAs appear in the Security Manager Administration window.



- 3 Right-click the cross-certificate that you want to remove from the directory, and then select **Remove from Directory**.
- 4 If prompted, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

When the cross-certificate you requested is removed from the directory, a dialog box appears confirming successful completion of the operation.

**5** Click **OK**.

You have now removed the cross-certificate you requested from the directory. The status is updated in the **Published in Directory** column. If you want to post the cross-certificate back to the directory later, see [“Publishing cross-certificates to the directory” on page 499](#).

# Publishing cross-certificates to the directory

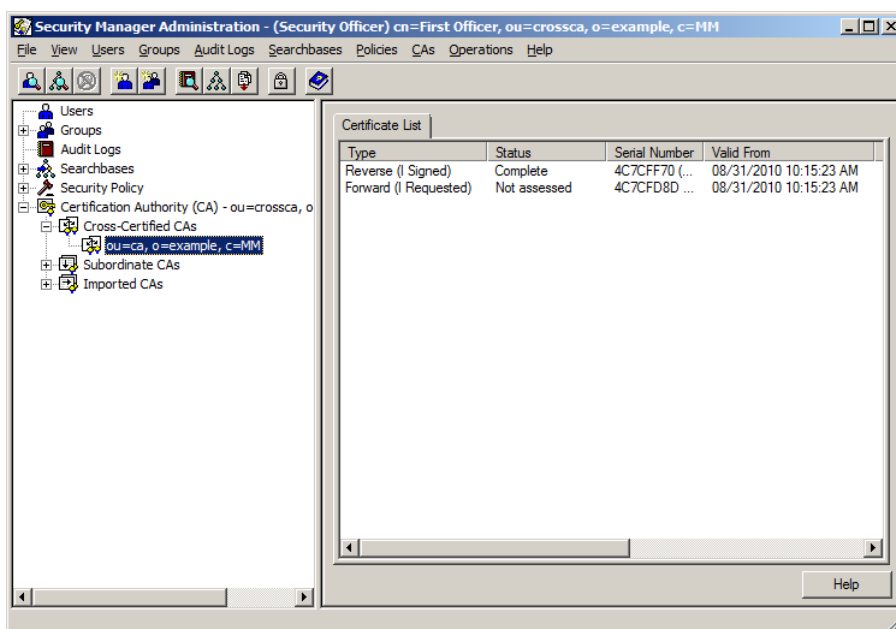
When you import a forward cross-certificate, Security Manager automatically saves the cross-certificate in the Security Manager database and in the directory. When your CA issues a reverse cross-certificate, Security Manager automatically saves the cross-certificate in the database and directory.

If you removed a cross-certificate from the directory (see [“Removing cross-certificates from the directory” on page 497](#)), you can post the cross-certificate back to the directory.

## To publish a cross-certificate to the directory

- 1 Log in to Security Manager Administration as a Security Officer or an Entrust PKI administrator with appropriate permissions.
- 2 In the tree view, expand **Certification Authority > Cross-Certified CAs** and then select DN of the other CA.

The cross-certified CAs appear in the Security Manager Administration window.



- 3 Right-click the certificate that you want to add to the directory, and click **Publish to Directory** in the pop-up menu.
- 4 If prompted, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

When the cross-certificate is added to the directory, a dialog box appears confirming successful completion of the operation.

**5** Click **OK**.

You have now added the cross-certificate you requested to the directory. The status is updated in the **Published in Directory** column.

# Revoking cross-certificates

Revoking a cross-certificate disables the extended third-party trust (or with unilateral cross-certification, one-way trust) previously established between two CAs. For example, suppose that Company One and Company Two mutually cross-certify. If Company One revokes the cross-certificate it signed for Company Two, users who belong to Company One can no longer encrypt data for and verify data from users who belong to Company Two. However, users in Company Two can still encrypt data for and verify data from users who belong to Company One.

You can only revoke reverse cross-certificates (cross-certificates signed by your CA). You cannot revoke forward cross-certificates (cross-certificates signed by another CA). You revoke a reverse cross-certificate for any of the following reasons:

- You no longer trust the other CA.
- Users in your CA no longer need to exchange secured files with users in the other CA.
- If a cross-certificate is about to expire and you want to cross-certify the CA again before the cross-certificate expires, you may want to revoke the cross-certificate before you cross-certify the other CA again.

You can also temporarily suspend the certificate by revoking it into an on hold state. Later, you can take the suspended certificate off hold, or revoke it completely.

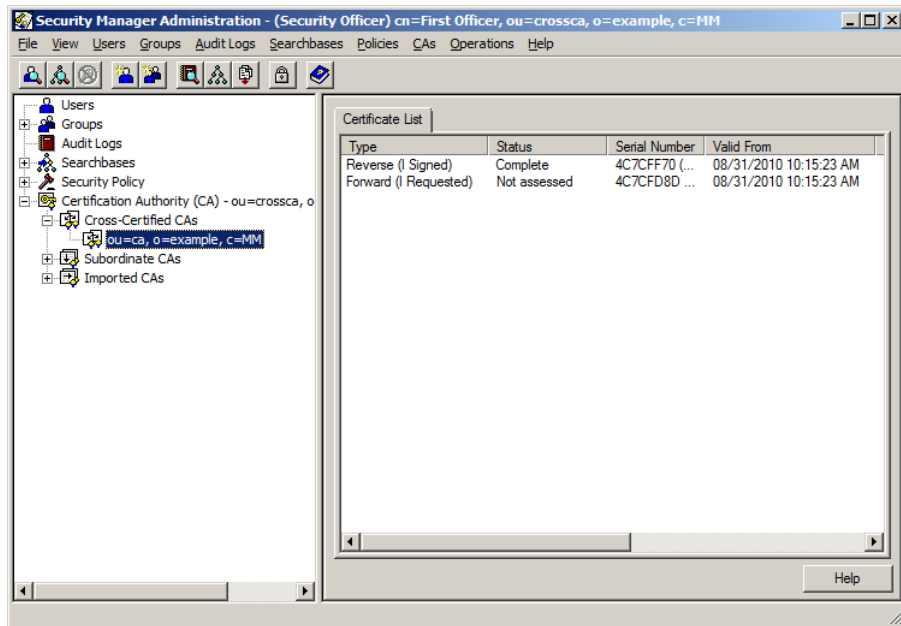
Revoked cross-certificates are referenced in an authority revocation list (ARL) stored in the directory. When users verify a cross-certified CA signed certificate, the Security Manager client application obtains a copy of the ARL from the directory. By default, ARLs update every six hours, but update immediately when a cross-certificate is revoked to ensure there is no delay before users become aware of a revoked cross-certificate. (With the exception of when the ARL is cached at the local client application. In this case, the application does not update its ARL until its cached ARL expires.)

Revoking a cross-certificate removes it from the directory. If you want to publish a revoked cross-certificate back into the directory, see [“Publishing cross-certificates to the directory” on page 499](#).

## To revoke a reverse cross-certificate

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 In the tree view, expand **Certification Authority > Cross-Certified CAs** and then select the DN of the other CA.

A list of forward and reverse cross-certificates for that CA appear in the Security Manager Administration window.



**3** Right-click the reverse certificate you want to revoke and then select one of the following options:

- To put the cross-certificate on hold, select **Revoke this certificate > On Hold**. Putting the cross-certificate on hold temporarily suspends the cross-certificate instead of revoking the certificate completely. You can take the cross-certificate off hold later (see [“Taking cross-certificates off hold” on page 504](#)).
- To revoke the cross-certificate because it was superseded by a more recent cross-certificate, select **Revoke this certificate > Superseded**.
- To revoke the cross-certificate because the CA key was compromised, select **Revoke this certificate > CA Key Compromise**.
- To revoke the cross-certificate without providing a reason for the revocation, select **Revoke this certificate > Unspecified**.

**4** If prompted, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

A confirmation dialog box appears when the revocation is complete.

**5** Click **OK**.

You have now revoked a cross-certificate. The status is updated in the **Status** column.

If you suspended the cross-certificate (put the cross-certificate on hold), you can take the cross-certificate off hold later (see [“Taking cross-certificates off hold” on page 504](#)).

Revoking a cross-certificate removes it from the directory. If you want to publish a revoked cross-certificate back into the directory, see [“Publishing cross-certificates to the directory” on page 499](#).

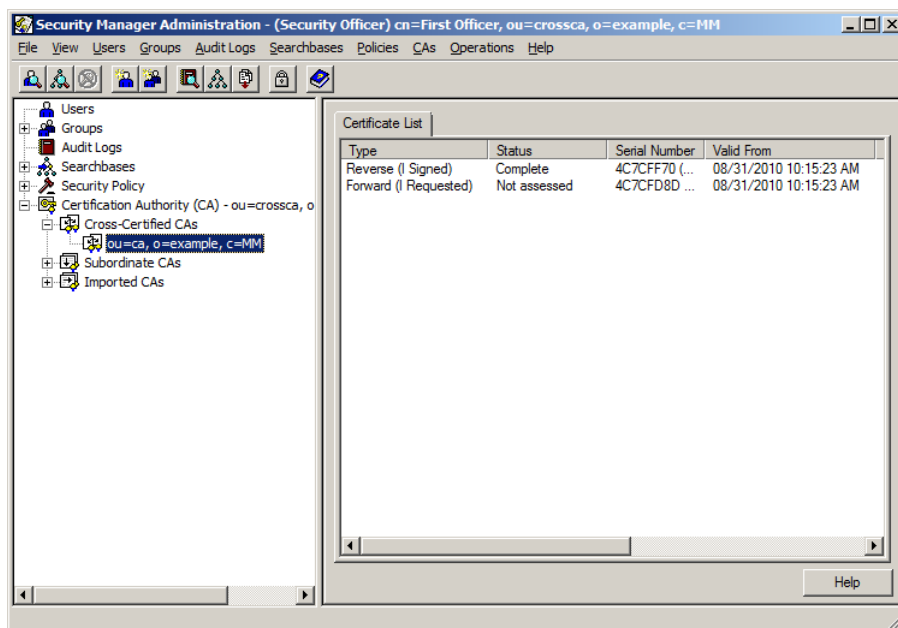
# Taking cross-certificates off hold

When revoking cross-certificates (see [“Revoking cross-certificates” on page 501](#)), you can suspend a cross-certificate (put the cross-certificate on hold) instead of revoking the cross-certificate completely. You can take a suspended cross-certificate off hold later. A cross-certificate taken off hold is published to the directory, if the cross-certificate was not published before the hold was canceled.

## To take a cross-certificate off hold

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 In the tree view, expand **Certification Authority > Cross-Certified CAs** and then select the DN of the other CA.

A list of forward and reverse cross-certificates for that CA appear in the Security Manager Administration window.



Cross-certificates on hold have a status of **Revoked: Certificate hold**.

- 3 Right-click the cross-certificate you want to take off hold and select **Cancel Hold**.
- 4 If prompted, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).  
A confirmation dialog box appears when the operation is complete.
- 5 Click **OK**.



# Updating cross-certificates

By default, cross-certificates are valid for 36 months. Administrators with sufficient permissions can change the lifetime of cross-certificates using the Administration Policy (see [“Configuring the Security Policy” on page 84](#)). The minimum lifetime for cross-certificates is two months, and the maximum lifetimes is 60 months.

Unlike users' certificates, cross-certificates are never automatically updated. Therefore, you must keep track of cross-certificate lifetimes and update the cross-certificates before they expire to ensure there is no interruption in service for users in the cross-certified CAs.

---

**Note:** When you update a reverse cross-certificate (a cross-certificate created and signed by your CA), the previous reverse cross-certificate is automatically revoked. If you are cross-certifying online, you must manually revoke the previous cross-certificate first (see [“Revoking cross-certificates” on page 501](#)). If you do not want Security Manager to automatically revoke the previous reverse cross-certificate, a Master User can set the advanced setting `RevokeCrossCert` to 0 (see the *Security Manager Operations Guide*).

---

## To update cross-certificates

Follow the steps in the appropriate section:

- If you are performing mutual or unilateral online cross-certification, see [“Performing online cross-certification” on page 475](#).
- If you are performing mutual or unilateral offline cross-certification, see [“Performing offline cross-certification” on page 484](#).

You have now updated the cross-certificates.

# Setting policy constraints requirements in protocol certificates

To secure the PKIX-CMP messages, Security Manager uses various protocol and encryption certificates. If you define policy constraints for your cross-certificates, clients may encounter errors when validating these certificates during key management. The protocol certificates do not include the required policy OIDs in the certificatePolicy extension to meet policy constraint requirements.

For example, you may see the following errors in an Entrust TruePass client log file:

```
[2004-05-07 20:54:59-0400][FATAL][AuthenticationProfileAccessServlet]
[TruePassHttpServlet.init] Unable to login administrator (Reporting Error:
TruePass: TPUser - Cannot access the user's crypto key
    Line number: 371
    File name: CurrentUser.cpp
    caused by EtkUser: EtkUser - The key update operation failed.
    caused by EtkUser: EtkMasterIdentity - The user certificate for securing
CA communications is revoked or has expired.
    caused by EtkCore: EtkCertificatePath - Require explicit policy is set,
and the authority constrained policy set became null.
com.entrust.truepass.javasecuritylibrary.JavaSecurityLibraryException: Reporting
Error:
TruePass: TPUser - Cannot access the user's crypto key
    Line number: 371
    File name: CurrentUser.cpp
    caused by EtkUser: EtkUser - The key update operation failed.
    caused by EtkUser: EtkMasterIdentity - The user certificate for securing
CA communications is revoked or has expired.
    caused by EtkCore: EtkCertificatePath - Require explicit policy is set,
and the authority constrained policy set became null.
```

The following procedure provides instructions for adding the required OIDs. It requires you to define the certificatePolicy extension for the verification and encryption certificates by defining their certificate types in the certificate specifications file.

## To set policy constraints requirements in protocol certificates

- 1 Export the certificate specifications file (see [“Creating the master.certspec file” on page 541](#)).
- 2 Scroll to the end of the [Certificate Types] section.

- 3 Add a certificate type for one or more of the following, depending on the requirements of your clients:
  - `ent_protocert`: to define the certificatePolicy for the protocol verification and encryption certificates
  - `ent_eventcert`: to define the certificatePolicy for the event server verification certificate

For example, to add both:

```

;
-----
; PKIX-CMP protocol verification and encryption Certificate Type
;
-----
ent_protocert=xcert,PKIX-CMP protocol,PKIX-CMP protocol
_continue_=certificates
;
-----
; PKIX-CMP protocol event server verification Certificate Type
;
-----
ent_eventcert=xcert,PKIX-CMP Event protocol,PKIX-CMP Event Server
_continue_=certificate

```

- 4 Scroll to the end of the [Extension Definitions] section.

- 5 Add the following information:

```

[ent_protocert Common Extensions]
certificatePolicies=2.5.29.32,n,m,DER,<ExtnValue>
[ent_eventcert Common Extensions]
certificatePolicies=2.5.29.32,n,m,DER,<ExtnValue>

```

where <ExtnValue> is the DER-encoded value for the OID. For example, 300830060604551D2000 is the DER-encoded value for the any-policy OID:

```

[ent_eventcert Common Extensions]
certificatePolicies=2.5.29.32,n,m,DER,300830060604551D2000

```

- 6 Save your changes.
- 7 Import the certificate specifications file (see [“Processing changes to the master.certspec file” on page 543](#)).
- 8 The certificates will automatically roll over in 24 hours. To reissue the certificates immediately, log in to the Security Manager Control Command Shell and run the `util proto-cert issue` and `util event-cert issue` commands. See the *Security Manager Operations Guide* for details.

- 9 (Optional.) To ensure that the certificatePolicy definition is correct, export the certificates and view them in a certificate viewer. See the *Security Manager Operations Guide* for details about exporting these certificates.

## Administering subordinate CAs

A hierarchy of CAs is a way of arranging two or more CAs in a restrictive trust relationship. In a hierarchy, a CA at any level signs the CA certificates of the CAs immediately below it in the hierarchy. Trust derives from a single trust anchor—called the root CA—at the top of the hierarchy. The other CAs in the hierarchy are known as subordinate CAs, and the terms subordinate and superior are used to distinguish relative position in the hierarchy.

For basic information about hierarchies, see the *Security Manager Deployment Guide*.

---

**Attention:** The subordinate CA's certificates are not standard CA certificates. They are issued as cross-certificates. If you need to edit the certificate extensions included in the subordinate CA certificate, edit the default cross-certificate type in the `master.certspec` or `initial.certspec` files on the root CA.

---

This chapter contains the following sections:

- [“Creating subordinate CA certificates” on page 510](#)
- [“Revoking subordinate CA certificates” on page 518](#)
- [“Taking subordinate CA certificates off hold” on page 520](#)
- [“Exporting subordinate CA certificates” on page 521](#)
- [“Preparing the directory” on page 523](#)

# Creating subordinate CA certificates

A superior CA creates subordinate CA certificates by processing certificate requests sent from subordinate CAs.

When initializing a subordinate CA or performing a subordinate CA key update, the subordinate CA must generate a new CA key and a certificate request. The superior CA must process the certificate request to create the new subordinate CA certificate. The subordinate CA must then import the subordinate CA certificate to complete the initialization or key update.

Using Security Manager Administration, your CA can process subordinate CA certificate requests sent from subordinate CAs using an online process or an offline process. In an online process, your CA uses a network connection to communicate directly with the subordinate CA to establish trust. In an offline process, you establish trust between your CA and the subordinate CA manually by exchanging request and response files.

---

**Note:** Before you add a subordinate CA, you must ensure that the subordinate and superior CAs can access each other's directory entries. See [“Preparing the directory” on page 523](#).

---

This section contains the following procedures:

- [“Creating subordinate CA certificates online” on page 510](#)
- [“Creating subordinate CA certificates offline” on page 513](#)

## Creating subordinate CA certificates online

A superior CA creates subordinate CA certificates by processing certificate requests sent from subordinate CAs. In an online process, your CA uses a network connection to communicate directly with the subordinate CA to establish trust.

You can create a subordinate CA certificate online only if you are initializing a Security Manager CA as a subordinate CA. If the subordinate CA is performing a key update, or if you are initializing a third-party CA as a subordinate CA, you must create the subordinate CA certificate using an offline process (see [“Creating subordinate CA certificates offline” on page 513](#)).

### To create a subordinate CA certificate online

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Select **CAs > Subordinate CAs > Begin creation of a Subordinate CA > Online**. The **Begin creation of a Subordinate CA** dialog box appears.

- 3 In the **Please enter the full DN of the subordinate CA** field, enter the full distinguished name (DN) of the subordinate CA. For example, `dc=Company Two, dc=com`.
- 4 In the **Certificate information** pane, choose a type of certificate to issue to the subordinate CA.  
 (You cannot change the certificate category, because technically, subordinate CA certificates are a kind of cross-certificate.) If you want to create subordinate CAs that have customized certificates, you must define new cross-certificate types in the `master.certspec` file. For more information on defining cross-certificate types, see [“Customizing cross-certificates” on page 580](#).
- 5 In the **Lifetime of CA** pane, set the lifetime of the subordinate CA certificate.  
 If you accept **Use default lifetime**, Security Manager uses the default value from the security policy. Alternatively, click **Use default lifetime** to deselect it, then select **Expires on** and enter the date (in the form mm, dd, yyyy) on which you want the certificate to expire, or **Expires in (months)** and enter the number of months for which you want the certificate to remain valid.

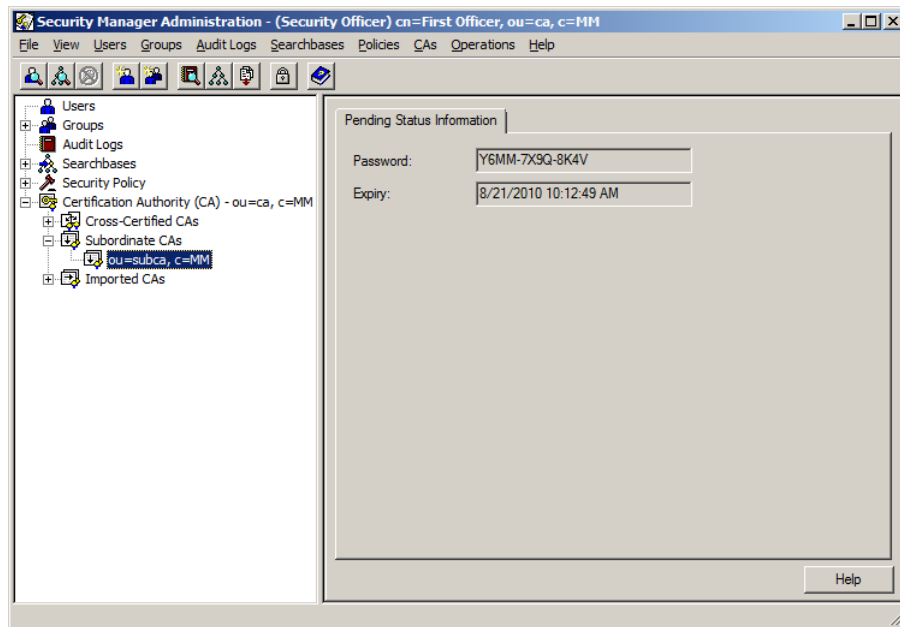
---

**Attention:** By default, Subordinate CA certificates are valid for 120 months (see [“Configuring the Security Policy” on page 84](#)). When the subordinate CA's certificates expire, the CA ceases to be operational. The subordinate CA's keys do not update. You should chose the lifetime of the subordinate CA certificate accordingly.

---

- 6 Click **OK**.
- 7 If prompted, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).
- 8 In the tree view, expand **Certification Authority (CA) > Subordinate CAs**, and then select the DN of the subordinate CA you just added.

The **Pending Status Information** property page shows the password required to initialize the subordinate CA, and the password expiry date. Note that the password expires in three days. If you do not initialize the subordinate CA with this password before the password expires, you must cancel the operation and start again.



- 9 Record the following:
  - the password
  - your CA DN
  - the IP address or DNS name of the server hosting Security Manager
  - the PKIX-CMP port (typically 829)



- your CA signing algorithm

Give this information securely to the person who is responsible for installing the subordinate CA. This information is needed when installing, configuring, and initializing the subordinate CA.

#### **10** Install the subordinate CA.

If the subordinate CA is a Security Manager CA, see the *Security Manager Installation Guide* for details.

You have now installed a subordinate in an online process. To view the completed operation select **CAs > Subordinate CAs > Refresh**.

## **Creating subordinate CA certificates offline**

A superior CA creates subordinate CA certificates by processing certificate requests sent from subordinate CAs. In an offline process, you establish trust between your CA and the subordinate CA manually by exchanging request and response files.

When initializing a subordinate CA or performing a subordinate CA key update, the subordinate CA must generate a new CA key and a PKCS#10 certificate request file. The superior CA must process the certificate request to create the new subordinate CA certificate. The subordinate CA must then import the subordinate CA certificate to complete the initialization or key update.

To create subordinate CA certificates offline, you must obtain a PKCS#10 file and a verification string from the subordinate CA. The PKCS#10 file contains the subordinate CA certificate request. You need the PKCS#10 file and verification string to process the request and generate the new subordinate CA certificate.

When processing the subordinate CA certificate request, Security Manager will create the new subordinate CA certificate and write it to a PKCS#7 file, and generate a verification string. The subordinate CA needs the PKCS#7 file and verification string to import the subordinate CA certificate.

### **To create a subordinate CA certificate offline**

#### **1** Install the subordinate CA.

If the subordinate CA is a Security Manager CA, see the *Security Manager Installation Guide* for installation instructions.

#### **2** At the subordinate CA, generate a PKCS #10 certificate request.

If the subordinate CA is a Security Manager CA, see the *Security Manager Installation Guide* for instructions. If you are adding a third-party subordinate CA, consult your third-party CA's documentation for instructions on creating this request.

Transfer the file from the subordinate CA to your CA.

- 3 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration”](#) on page 46).
- 4 Select **CAs > Subordinate CAs > Begin creation of a subordinate CA > Offline**. The **Create subordinate CA offline** dialog box appears.

**Create subordinate CA offline**

Location of PKCS#10 request

Enter the path and filename of a PKCS#10 certificate request. Obtain this file from the administrator of the subordinate CA you want to add.

Destination and format of response

Enter a path and filename for the response to the PKCS#10 request. Also, choose the content and format of the response. Give the response file to the administrator of the subordinate CA.

Which certificates would you like to include in the response?

☐ Subordinate CA certificates only, in plain format.

☒ All certificates in the certificate chain, in PKCS#7 format.

Which format would you like to save the response as?

☒ Binary

☐ Base 64 (PEM encoded)

- a In the **Location of PKCS #10 request** field, enter the path and name of the PKCS #10 request file, or click **Browse** to locate the file.
- b In the **Destination and format of response** field, enter the path and name of the response file that you will create, or click **Browse** to choose a location.  
A response file contains the subordinate CA certificates signed by the superior CA.
- c In the **Which certificates would you like to include in the response?** pane:
  - To include only the subordinate CA certificates in the response file, select **Subordinate CA certificates only, in plaintext format**.  
Select this option if you are adding a third-party subordinate CA and the subordinate CA requires only the subordinate CA certificates in the response file.
  - To include the subordinate CA certificates and all superior CA certificates (up to and including the root CA certificates) in the response file, select **All certificates in the certificate chain, in PKCS #7 format**.

Select this option if you are adding an Entrust subordinate CA.

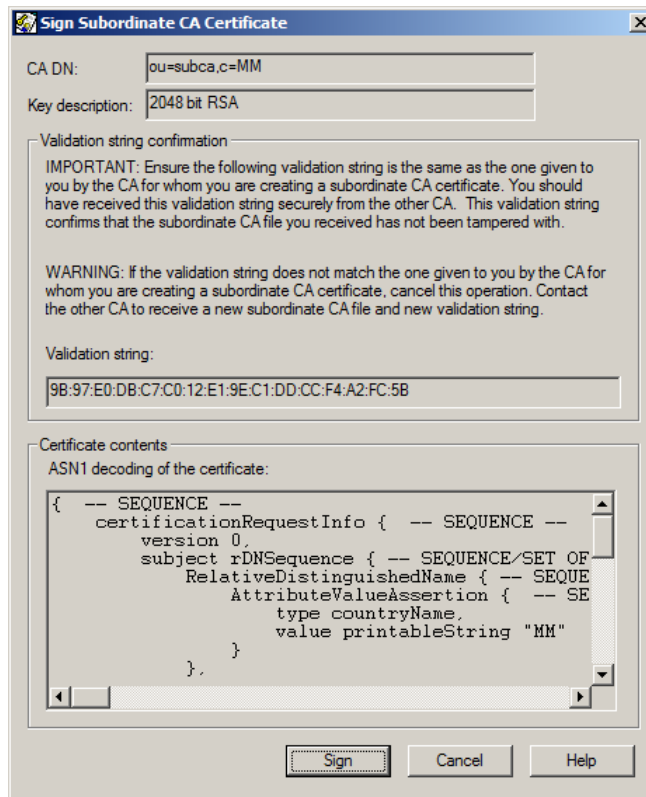
**d** In the **Which format would you like to save the response as?** pane:

- To save the response file in binary format, select **Binary**. Security Manager Administration will append a `.der` extension to the file name.
- To save the response file in Base 64 (PEM) format, select **Base 64 (PEM encoded)**. Security Manager Administration will append a `.pem` extension to the file name.

The choice of formats is provided for interoperability with PKIs from other vendors. Security Manager can read files in either format.

**5** Click **OK**.

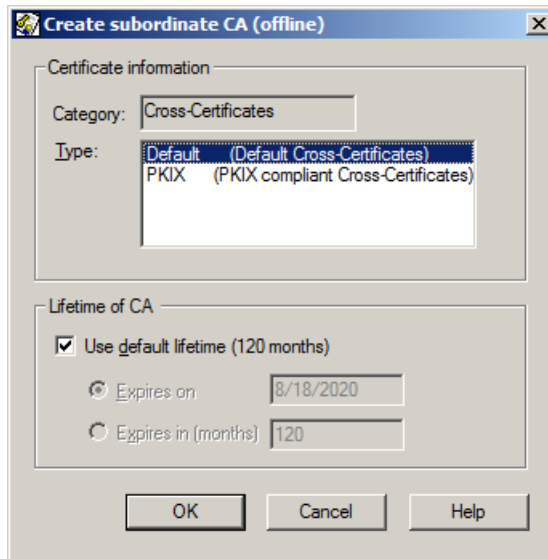
The **Sign Subordinate CA Certificate** dialog box appears.



**6** Use the validation string to validate the PKCS #10 request before signing it. Compare this string to the validation string provided by the administrator at the subordinate CA.

**7** Click **Sign**.

The **Create subordinate CA (offline)** dialog box appears.



- a** In the **Type** field, select the type of certificate.
- b** In the **Lifetime of CA** group box, set the lifetime of the subordinate CA certificate.

To use the default lifetime from the security policy (by default, 120 months; see [“Configuring the Security Policy” on page 84](#)), select **Use default lifetime**. By default, this option is selected.

To select a custom lifetime, deselect **Use default lifetime** and then select one of the following options:

- To enter an expire date, select **Expires on** and then enter a the expire date in the form of MM/DD/YYYY.
- To enter a lifetime (in months), select **Expires in (months)** and then enter a lifetime (in months).

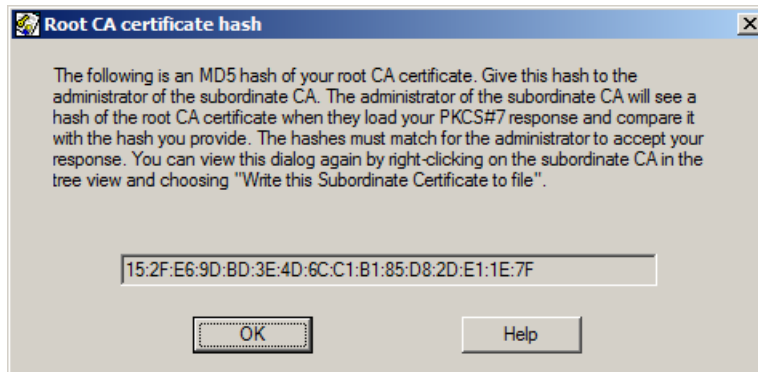
---

**Attention:** When the subordinate CA's certificates expire, the CA will cease to be operational. The subordinate CA's keys will not be updated. The lifetime of the subordinate CA certificate should be chosen accordingly.

---

- 8** Click **OK**.
- 9** If prompted, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

The **Root CA certificate hash** dialog box appears.



- 10** Record this value and securely give it to the subordinate CA administrator.

The subordinate CA administrator needs this string to confirm that you sent the response file and that it is unaltered.

---

**Attention:** Give your validation string to the administrator securely and separately from your response file. Sending the response and its validation string together greatly increases the risk of undetectable tampering with the response.

---

- 11** Click **OK**.

- 12** Transfer the response file to the subordinate CA.

- 13** At the subordinate CA, import the response file.

If the subordinate CA is a Security Manager CA, see the Security Manager documentation for instructions. If the subordinate CA is a third-party CA, consult the documentation for your subordinate CA for instructions.

You have now added a subordinate CA in an offline process.

# Revoking subordinate CA certificates

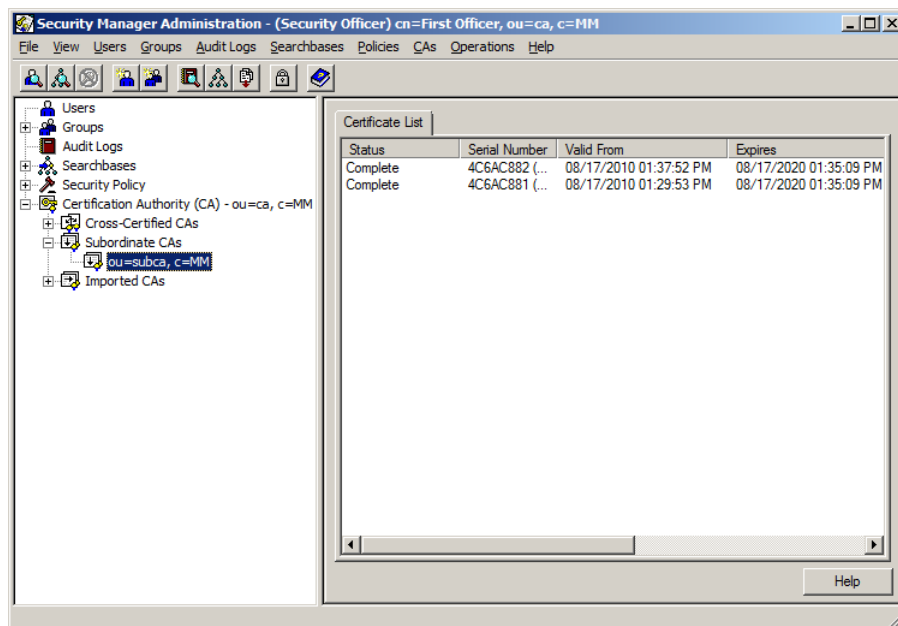
If a subordinate CA's signing key pair is compromised, you should revoke the certificate. You can also temporarily suspend a subordinate CA by revoking it into an on hold state. Later, you can take the suspended certificate off hold, or revoke it completely.

Revoking the subordinate CA certificate is a very serious operation and should be carried out with extreme caution. If you decide to revoke the certificate, any user certificates signed by that CA before it was revoked are no longer valid. After you issue a new certificate for the subordinate CA (see [“Creating subordinate CA certificates” on page 510](#)), you must recover all the users of that CA. For information about recovering users, see [“Recovering user key pairs” on page 162](#). A Master User can also recover all users with the Security Manager Control Command Shell (see the *Security Manager Operations Guide*).

## To revoke a subordinate CA certificate

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Click **Certification Authority (CA)** in the tree view of Security Manager Administration to expand it, and click **Subordinate CAs** to expand it.
- 3 Click the subordinate CA whose certificate you want to revoke.

The **Certificate List** property page shows all certificates issued to this subordinate.



- 4 On the **Certificate List** property page, right-click the certificate that you want to revoke. In the pop-up menu, choose **Revoke this certificate** and one of the following revocation reasons:
  - Click **On Hold** if you want to suspend the activity of the subordinate CA on a temporary basis. From this state, you can later take the subordinate CA's certificate off hold (see ["Taking subordinate CA certificates off hold" on page 520](#)), or you can revoke the certificate of the subordinate CA.
  - Click **Superseded** when you do not trust the certificate, but you know the key is not compromised. For example, choose **Superseded** when you are revoking a subordinate because its PKI administrators have contravened your certificate practice statements.
  - Click **CA Key Compromise** when the subordinate's signing key pair is compromised.
  - Click **Unspecified** when the other reasons do not apply.
- 5 Authorize the operation. For details on authorizing operations, see ["Authorizing sensitive operations" on page 52](#).

You have now revoked a subordinate CA certificate.

If you want to issue a new certificate for the subordinate, see ["Creating subordinate CA certificates" on page 510](#).

# Taking subordinate CA certificates off hold

When revoking subordinate CA certificates (see [“Revoking subordinate CA certificates” on page 518](#)), you can suspend a subordinate CA certificate (put the certificate on hold) instead of revoking the certificate completely. You can take a suspended certificate off hold later.

## To take a subordinate CA certificate off hold

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** In the tree view, expand **Certification Authority (CA) > Subordinate CAs**.
- 3** Select the subordinate CA whose certificate you want to take off hold.
- 4** Right-click the subordinate CA certificate and select **Cancel Hold**.

You have now taken the revoked subordinate CA certificate off hold and made it trusted again.



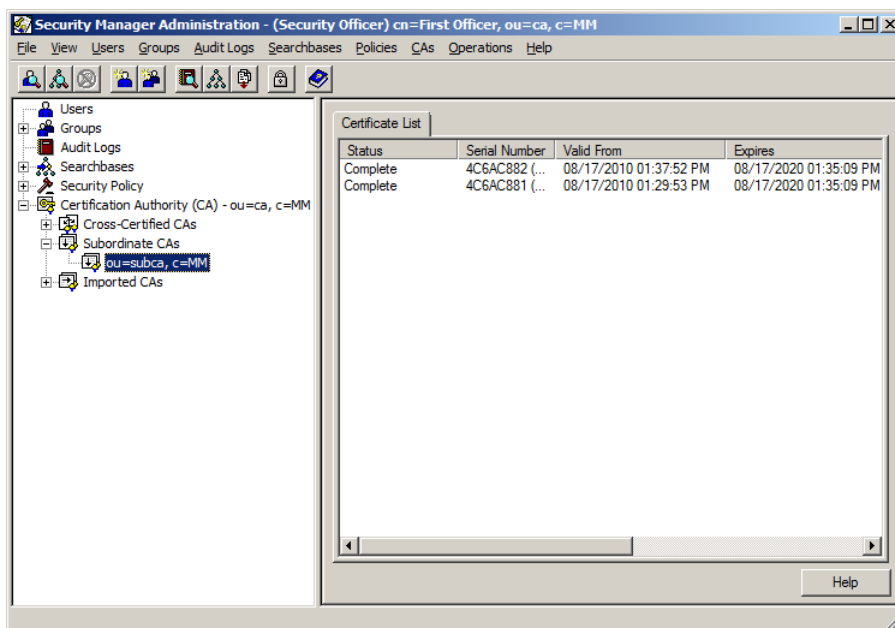
# Exporting subordinate CA certificates

Some applications may require a CA certificate from other applications as a prerequisite for working with them. If an application you want Security Manager to work with has this requirement, you can export the subordinate CA's certificate to a file and submit the file to the other application. You can also export the subordinate CA certificate if you lose it during the initialization or key update process.

## To export a subordinate CA certificate

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration”](#) on page 46).
- 2 In the tree view, expand **Certification Authority (CA) > Subordinate CAs** and then select the distinguished name of the subordinate CA.

The **Certificate List** property page shows all certificates issued to this subordinate.

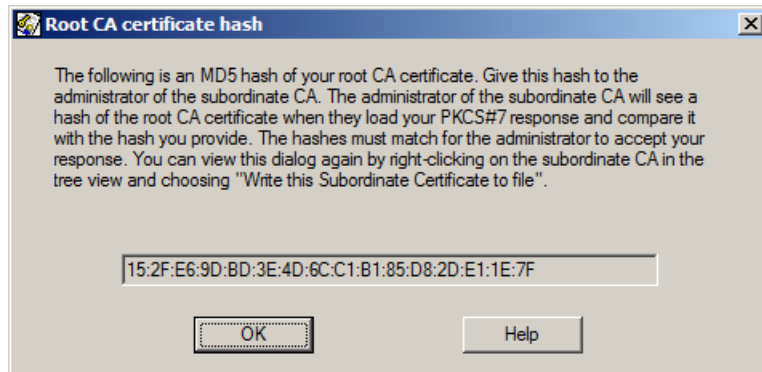


- 3 Right-click the certificate that you want to save to a file, and then select **Write this Subordinate Certificate to file**.

The **Write Subordinate CA Certificate to File** dialog box appears.

- 4 Select a location and enter a name for the file and then click **Save**.

The **Root CA certificate hash** dialog box appears.



**5** Record the validation string. You need this validation string if you need to import the subordinate CA into an application.

**6** Click **OK**.

You have now saved a subordinate CA's certificate to file.

# Preparing the directory

CAs in a hierarchy (and cross-certified CAs) need to access each other's directory entries. These entries include certificate revocation lists (CRLs), authority revocation lists (ARLs), and encryption certificates. Preparing the directory means ensuring that CAs in a hierarchy can access each other's directory entries.

When you plan a hierarchy of CAs, you must decide how many directories you plan to install. For example, when you install Security Manager, you have the option of installing a local directory or using an existing directory.

CAs that share a directory can form a hierarchy or cross-certify because they can access each other's directory entries. CAs that do not share a directory cannot access each other's directory entries (and therefore cannot form a hierarchy or cross-certify) unless you chain or use LDAP referrals to connect their directories.

## Chaining and referring directories

If you choose to install one or more local directories for CAs in your hierarchy, you must configure them so that the CAs can search across them all seamlessly for entries. One way to configure these directories is by chaining them to each other.

To understand chaining, suppose directory A is chained to directory B, and a client requests data from the Directory Service Agent (DSA) for directory A. If DSA A cannot find the requested data in directory A, it passes the client's request to DSA B. If DSA B finds the data, it returns it to DSA A, which in turn sends it to the client. If directory B is chained back to directory A (called two-way chaining), then client requests to DSA B are similarly sent to DSA A.

Another way to configure directories so that CAs can search across them all is to refer the directories to each other. To understand referring, suppose directory A is referred to directory B, and a client requests data from the Directory Service Agent (DSA) for directory A. If DSA A cannot find the requested data in directory A, it tells the client to request the data from DSA B. In other words, DSA A refers clients to DSA B if it cannot fulfill requests locally. If directory B is referred back to directory A (called two-way referring), then DSA B can also refer client requests to DSA A. Referring is less efficient than chaining.

For more information on chaining and referring directories, consult your directory vendor's documentation.



## Customizing certificates

Security Manager certificates contain keys generated by Security Manager and other information, such as the date that the certificate expires. Security Manager makes use of many different kinds of certificates, including Enterprise, Web, cross-certificates, policy, and CA certificates. If your role includes the appropriate permissions (see [“Permissions reference” on page 371](#)), you can customize certificates.

---

**Attention:** The default certificates meet the requirements of most organizations. Only a highly trusted system administrator with advanced technical knowledge should perform the procedures described in this chapter for customizing certificates.

---

This chapter does not describe how to configure CRL distribution points (CDPs). For details about configuring CDPs, see the *Security Manager Operations Guide*.

This chapter includes the following sections:

- [“About certificates” on page 527](#)
- [“Predefined certificate types” on page 535](#)
- [“Working with the master.certspec file” on page 539](#)
- [“Customizing Enterprise and Web certificates” on page 544](#)
- [“Customizing policy certificates” on page 571](#)
- [“Customizing cross-certificates” on page 580](#)
- [“Customizing CA certificates” on page 591](#)
- [“Excluding certificate extensions by certificate type” on page 593](#)
- [“Making a certificate type obsolete” on page 599](#)
- [“Customizing database fields” on page 601](#)

- “Turning off revocation checking for foreign certificates” on page 605

# About certificates

Security Manager certificates contain keys generated by Security Manager and other information, such as the date that the certificate expires. Security Manager certificates are issued by Security Manager from several certificate categories.

Typically, you customize certificates to meet the certificate requirements of your organization while ensuring the authenticity of the certificate content. You can customize certificates by defining extra content and by excluding default content. Extra content may be of a fixed or variable nature. For example, you can use variable content to customize certificates on a per-user basis. You can specify that a certificate issued to a user must include a value that represents the expenditures the user is allowed to make on behalf of the company (for example, under \$500.00, under \$2000.00, and so on).

You can also specify that you no longer want a particular type of certificate issued, and that you want content about users added to the Security Manager database rather than to a certificate.

Topics in this section:

- [“Security Manager certificate categories” on page 527](#)
- [“About the X.509 recommendation” on page 528](#)
- [“Dividing certificate categories into certificate types” on page 529](#)
- [“Why customize certificates?” on page 530](#)
- [“Working with Security Manager certificate specifications” on page 531](#)

## Security Manager certificate categories

Security Manager certificate categories determine the purpose of certificates issued from those categories.

Table 46 lists the Security Manager certificate categories and the purpose of certificates issued from those categories.

**Table 46:** Security Manager certificate categories

Certificate category	Purpose of certificates
Enterprise	These certificates identify the owner of the certificate (the user Security Manager issued the certificate to). The owner is an Enterprise user.
Web	These certificates identify the owner of the certificate (the user Security Manager issued the certificate to). The owner is a Web user.
cross-certificate	These certificates establish direct trust between CAs, allowing the users who belong to each CA to indirectly trust each other.
policy	These certificates define policy settings for users.

**Table 46:** Security Manager certificate categories (continued)

Certificate category	Purpose of certificates
CA	These certificates store the verification public key of the CA's signing key pair.

Certificates from different categories can be used for different purposes because of the content that Security Manager inserts by default when issuing certificates. This default content is determined by the X.509 recommendation.

## About the X.509 recommendation

The X.509 recommendation is a generally accepted definition of standard certificate content within the certification industry. Certificates that follow the X.509 recommendation are called X.509 compliant.

The X.509 recommendation is maintained and published by the International Telecommunications Union (ITU). For information about this recommendation, contact the ITU and request *ITU-T Recommendation X.509*.

- The first edition of the X.509 standard was published in 1988 and the certificate format version was defined as v1.
- The second edition of the X.509 standard was published in 1993 and the certificate format version was defined as v2. This version was enhanced by the addition of the new fields, `issuerUniqueID` and `subjectUniqueID`.
- The third edition of the X.509 standard was published in 1997 and the certificate version was defined as v3. This version was enhanced by the addition of the certificate extensions field, a generic extensions field specifically added to eliminate future certificate format revisions.
- The fourth edition of the X.509 standard was published in 2000/2001 and the fifth edition in 2005/2006. Both editions of the standard only define new extensions that fit within the certificate extensions field defined by the version 3 certificate format defined in the 3rd edition of the standard. Therefore, the fourth and fifth editions use the same X.509 v3 certificate format.

Security Manager certificates include extensions and therefore comply with the version 3 certificate format, used in the third, fourth, and fifth editions of the X.509 standard. From a functional perspective, Security Manager complies with all editions of the X.509 standard.

Security Manager supports most standard extensions; however, you may want to review extensions for specific conformance requirements with Customer Support.



---

**Note:** This chapter assumes that you have a basic working knowledge of the X.509 recommendation. If you are not familiar with this recommendation, do not attempt to perform the procedures in this chapter.

---

Each certificate category contains one or more types of certificates. You use certificate types to customize certificates (for example, to define content for Security Manager to insert in certificates, in addition to the X.509 content it inserts by default).

## Dividing certificate categories into certificate types

Each certificate category contains one or more certificate types. Table 47 lists the certificate categories and names their default certificate types. These default certificate types cannot be made obsolete, although you can change their name and description.

**Table 47:** Certificate categories

This certificate category...	includes default type(s) named...
Enterprise	Default
Web	Default
cross-certificate	Default
policy	Client settings
CA	Default, Link

Certificates issued from a certificate category are based on a particular certificate type. A certificate type determines how Security Manager customizes certificates issued for that type. Certificate types are defined using Security Manager certificate specifications.

---

**Note:** The certificate categories include at least one default certificate type. This means that you are not required to define or modify any certificate types using certificate specifications.

---

In addition to the default certificate types, the `master.certspec` file includes many other certificate types that are enabled by default. The following excerpt from the `master.certspec` file shows the certificate type descriptions for the Timestamp and Roaming Server certificate types:

```
[Certificate Categories]
```

```
...
```

```
[Certificate Types]
```

```

...
; -----
; Timestamp Certificate Type
; -----
ent_timestamp=enterprise, Timestamping Agent, Timestamping Agent Certifica
_continue=tes
;
; -----
; Roaming Server Certificate Type
; -----
ent_profsrvr=enterprise, Roaming Server, Roaming Server Certificates
...

```

If you do not use the product or feature associated with a certain certificate type, you can define these certificate types as obsolete in the `master.certspec` file, or comment them out before initialization in the `initial.certspec` file. For information about defining certificate types as obsolete, see [“Making a certificate type obsolete” on page 599](#).

## Why customize certificates?

Typically, you customize certificates to meet the certificate requirements of your organization while ensuring the authenticity of the certificate content. For example, you can customize certificates by adding information to them. Because the CA digitally signs certificates and you trust this signature, you can trust the integrity of the certificate contents (only an Entrust PKI administrator with the appropriate permissions can change the contents). For example, if you chose to add this information to the directory directly, you are not assured of its authenticity. Because this information is not signed, and there is no requirement to locate the directory on a physically secured server, anyone with access to the directory can change the unsigned information in it.

---

**Note:** Certificate content is digitally signed; it is not encrypted. Do not add content that requires encryption to certificates (for example, information such as employees' salaries that you should keep private). If this information is related to users, you can add it to the Security Manager database using database fields (see [“Customizing database fields” on page 601](#)).

---

## Customizing content for certificates

Security Manager certificate specifications allow you to define how Security Manager customizes certificates after inserting the other X.509 content.

You can work with certificate specifications to customize certificates for your organization. For example, suppose that you want the certificates that Security Manager issues to Enterprise users to contain the department within your organization that the certificate owner belongs to. Using certificate specifications, you

- identify the certificate type that you want to add this content to (in this example, the Default certificate type in the Enterprise certificate category)
- define the content (in this example, the department name) that Security Manager is to insert in certificates of this type

You can customize all the certificates that Security Manager issues. The customizations should reflect how the certificates are used. For example, you can add identifying information about users to certificates issued from the Enterprise and Web certificate categories. However, this information is not useful to add to certificates issued from the cross-certificate category because cross-certificates are not used to identify users.

## Working with Security Manager certificate specifications

The Security Manager database stores Security Manager certificate specifications. You can work with Security Manager certificate specifications using the Security Manager certificate specification file. This file is an ASCII (text) file.

This file is available before and after you initialize Security Manager. Before initialization, the file is named `initial.certspec` and is stored in the following location, by default, on the Security Manager server:

```
c:\Program Files\Entrust\Security Manager\etc\initial.certspec
```

---

**Note:** If you are running Security Manager in an Identrus environment, see the *Security Manager Operations Guide*.

---

This is the only time that the file exists without being explicitly created by a Security Officer. This file is made available to you so that you can customize certificates before first-time initialization. For example, Security Manager issues certificates for the CA and the First Officer during initialization. Using the `initial.certspec` file, you can customize even the very first certificates issued by Security Manager for your organization.

After initialization, by default the file is named `master.certspec` to distinguish it from the pre-initialized version. Note that this file does not exist until you export it

from Security Manager Administration. For more information, see [“Creating the master.certspec file” on page 541](#).

---

**Note:** For the remainder of this chapter, references to `master.certspec` apply as well to `initial.certspec` unless noted otherwise.

---

---

**Attention:** Do not use backslashes (\) in the certificate specification file. Doing so may cause errors. Instead, use a forward slash (/).

---

## Contents of the master.certspec file

This is an excerpt from the default `master.certspec` file.

```
[Certificate Categories]
;*****
;* This section lists the categories to which certificate types may be *
;* assigned. Note that the order in which they are listed is not      *
;* important.                                                         *
;*                                                                     *
;*                                                                     *
;* No changes to this section are required.                           *
;*****
1=enterprise
2=web
4=xcert
5=policycert
6=cacert
...
```

The default `master.certspec` file is compliant with the X.509 recommendation. This file also meets the minimum requirements of the Internet Engineering Task Force (IETF) Request for Comments (RFC) 3280: *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*.

The recommendations in RFC 3280 are based on the X.509 recommendation, but are more stringent. The default `master.certspec` file meets the minimum requirements of RFC 3280. It does not enforce the RFC 3280 suggestions for stricter requirements. However, you can edit the file to meet those suggestions if necessary.

You can work with the `initial.certspec` and `master.certspec` files if you are a Security Officer or if you have the appropriate permissions (see [“Administering roles” on page 353](#)). The steps to access and process Security Manager certificate

specifications are different for the `initial.certspec` and `master.certspec` files. The following steps provide a summary of the procedures for each file. These procedures are described in more detail in the remainder of this chapter.

Procedures in this section:

- [“To work with the initial.certspec file” on page 533](#)
- [“To work with the master.certspec file” on page 533](#)

### To work with the initial.certspec file

- 1 Open the `initial.certspec` file for editing (see [“Opening the master.certspec file” on page 541](#)).
- 2 Edit the `initial.certspec` file as described in the following sections:
  - [“Customizing Enterprise and Web certificates” on page 544](#)
  - [“Customizing policy certificates” on page 571](#)
  - [“Customizing cross-certificates” on page 580](#)
  - [“Customizing CA certificates” on page 591](#)
  - [“Making a certificate type obsolete” on page 599](#)
  - [“Customizing database fields” on page 601](#)

---

**Attention:** Do not use backslashes (\) in the certificate specification file. Doing so may cause errors. Instead, use a forward slash (/).

---

- 3 Initialize Security Manager (see the *Security Manager Installation Guide*).

During initialization, Security Manager writes the Security Manager certificate specifications in the `initial.certspec` file to the Security Manager database and creates a file named `initial.log`. If the specifications are not successfully written to the database, `initial.log` contains a list of errors that you must resolve before `initial.log` can process successfully.

### To work with the master.certspec file

- 1 Log in to Security Manager Administration as a Security Officer (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Create the `master.certspec` file as described in [“Creating the master.certspec file” on page 541](#).  
Security Manager Administration imports Security Manager certificate specifications from the Security Manager database.
- 3 Open the `master.certspec` file for editing (see [“Opening the master.certspec file” on page 541](#)).

**4** Edit the `master.certspec` file as described in the following sections:

- [“Customizing Enterprise and Web certificates” on page 544](#)
- [“Customizing policy certificates” on page 571](#)
- [“Customizing cross-certificates” on page 580](#)
- [“Customizing CA certificates” on page 591](#)
- [“Making a certificate type obsolete” on page 599](#)
- [“Customizing database fields” on page 601](#)

---

**Attention:** Do not use backslashes (\) in the certificate specification file. Doing so may cause errors. Instead, use a forward slash (/).

---

**5** Process the `master.certspec` file (see [“Processing changes to the master.certspec file” on page 543](#)).

Security Manager writes the Security Manager certificate specifications in the `master.certspec` file to the Security Manager database. Security Manager Administration then deletes the `master.certspec` file.

Once you write the contents of the `master.certspec` file into the Security Manager database, Security Manager begins using the revised Security Manager certificate specifications immediately.

# Predefined certificate types

Security Manager Administration divides certificates into two categories: Web and Enterprise. Each certificate category contains one or more certificate types. A certificate type determines how Security Manager customizes certificates issued for that type. Security Manager includes several predefined Enterprise and Web certificate types.

This section contains the following topics:

- [“Predefined Enterprise certificate types” on page 535](#)
- [“Predefined Web certificate types” on page 538](#)

## Predefined Enterprise certificate types

Table 48 describes the Enterprise certificate types.

**Table 48:** Predefined Security Manager Enterprise certificate types

Certificate Type (Certificate name)	Description
1-Key-Pair User (ent_skp_dualusage)	Certificate type that defines a dual-usage encryption and signing key pair for 1-key-pair users.
2-Key-Pair User (ent_twokeypair)	Certificate type that defines encryption and verification key pairs for 2-key-pair users.
Admin Services User Management (ent_admsrvcs_usrmgmt)	Certificate type for the User Management Service profile in Entrust Authority Administration Services.
Admin Services User Registration (ent_admsrvcs_usrreg)	Certificate type for the User Registration Service profile in Entrust Authority Administration Services.
Default (ent_default)	The default Enterprise certificate type. Security Manager uses this certificate type if a PKI administrator does not select another certificate type.
Desktop Admin (ent_desktop)	Certificate type for administrators responsible for deploying Entrust Entelligence Desktop Solutions software.
EFS User (ent_efs)	Certificate type that defines encryption, verification, and encryption file system (EFS) key pairs for 3-key-pair users.
Email Content Scanner (ent_msgscanner)	Certificate type for email content processors, such as virus scanners or email archivers.

**Table 48:** Predefined Security Manager Enterprise certificate types (continued)

Certificate Type (Certificate name)	Description
Enterprise Domain Controller (ent_ad_dc)	Certificate type for Microsoft Windows domain controllers. This domain controller certificate type is compatible with Entrust Entelligence Security Provider.
Enterprise Machine (ent_machine)	Certificate type for computer digital IDs. This certificate type is compatible with Entrust Entelligence Security Provider.
ePassport - Document Signer (epass_doc_signer)	Certificate type for signing the Document Security Object on electronic passports. This certificate type is issued by a Country Signing Certification Authority (CSCA) to a Document Signer.
ePassport- IS Attached Client (ent_eaccattached)	Certificate type for Entrust Authority IS Client profiles running in attached mode.
ePassport - IS Concentrator (ent_eaccon)	Certificate type for Entrust Authority IS Concentrator profiles.
ePassport - IS Standalone Client (ent_eaccstandalone)	Certificate type for Entrust Authority IS Client profiles running in standalone mode.
ePassport - Master List Signer (ent_mlist_signer)	Certificate type for providing a master list of trusted foreign Country Signing Certification Authorities (CSCAs). This certificate type is issued by a CSCA.
ePassport - Master List Signer Administrator (ent_mlist_admin)	Certificate type for Master List Signer administrators.
ePassport - SPOC Administrator (ent_spoc_admin)	Certificate type for Single Point of Contact (SPOC) administrators.
ePassport - SPOC Client (ent_spoc_client)	Certificate type for the SPOC Web Service client profiles in Entrust Authority Administration Services.
ePassport - SPOC DV Client (ent_spoc_dv)	Certificate type for SPOC DVCKM Client profiles in Entrust Authority Administration Services.
ePassport - SPOC Server (ent_spoc_server)	Certificate type for the SPOC Web Service server profiles in Entrust Authority Administration Services.



**Table 48:** Predefined Security Manager Enterprise certificate types (continued)

Certificate Type (Certificate name)	Description
Export (ent_export)	Certificate type that includes the certificate extension required by Entrust clients that allows them to export the corresponding private key.
IPSec Device (vpn_dir)	Certificate type used by Enrollment Server for VPN to create VPN server certificates that are stored in the directory.
IPSec Device (vpn_nodir)	Certificate type used by Enrollment Server for VPN to create VPN server certificates that are not stored in the directory.
Messaging Server (ent_msgserver)	Certificate type for Entrust Entelligence Messaging Server profiles.
MS VPN Client User (vpn_client_user)	Certificate type that defines a dual-usage key pair for Microsoft VPN client users.
Nonrepudiation User (ent_nonrepud)	Certificate type that defines encryption, verification, and nonrepudiation key pairs for 3-key-pair users.
Nonrepudiation/EFS User (ent_nonrepud_and_efs)	Certificate type that defines encryption, verification, nonrepudiation, and EFS key pairs for 4-key-pair users.
PKCS10 2-Key-Pair User (ent_twokeypair_p10)	Certificate type that defines verification and encryption key pairs for PKCS#10 2-key-pair users.
Roaming Server (ent_profsrvr)	Certificate type for Entrust Authority Roaming Server profiles.
Smart Card Logon for MS Security Framework Users (ent_ms_smrtcrd_capi)	Certificate type that defines a dual-usage key pair for Smart Card Logon for Microsoft Cryptographic API users.
Smart Card Logon for PKCS#11 Users (ent_msft_smartcard)	Certificate type that defines encryption and verification key pairs for Smart Card Logon for PKCS#11 users.
Standalone EFS User (ent_standalone_efs)	Certificate type that defines CMP signing and EFS key pairs for 2-key-pair users.
Timestamping Agent (ent_timestamp)	Certificate type for securing Timestamping Agents. The extended key usage extension is marked as non-critical.
Timestamping Agent Critical (ent_timestamping)	Certificate type for securing Timestamping Agents. The extended key usage extension is marked as critical.

**Table 48:** Predefined Security Manager Enterprise certificate types (continued)

Certificate Type (Certificate name)	Description
TruePass Server (ent_truepass)	Certificate type used for Entrust TruePass server profiles.
TruePass Server Multidomain Primary (ent_truepass_multi)	Certificate type used for the primary Entrust TruePass server profile in a multiple-domain environment.
XAP Server (ent_xapsrv)	Certificate type for the XAP server. The ASH profile uses this certificate type.

## Predefined Web certificate types

[Table 49 on page 538](#) describes the Web certificate types.

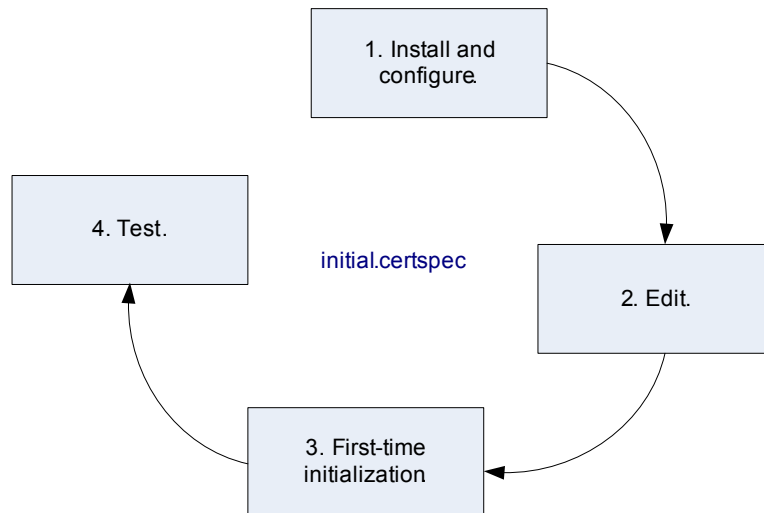
**Table 49:** Predefined Security Manager Web certificate types

Certificate Type (Certificate name)	Description
Code Signing (web_codesign)	Certificate type designed to verify code signed by Netscape or Microsoft code signing technologies.
Default (web_default)	<p>The default Enterprise certificate type. Security Manager uses this certificate type if a PKI administrator does not select another certificate type.</p> <p>Certificates issued using the default Web certificate type are enabled for S/MIME and SSL clients. This certificate type was designed to work with Entrust Authority Enrollment Server for Web.</p>
Domain Controller (web_ad_dc)	Certificate type for Microsoft Windows domain controllers. This domain controller certificate type is not compatible with Entrust Entelligence Security Provider.
MS VPN Client Machine (vpn_client_machine)	Certificate type for Microsoft VPN clients.
MS VPN Server (ms_vpn_server)	Certificate type for Microsoft VPN servers.
Web Server (web_server)	Certificate type for SSL Web servers.

# Working with the master.certspec file

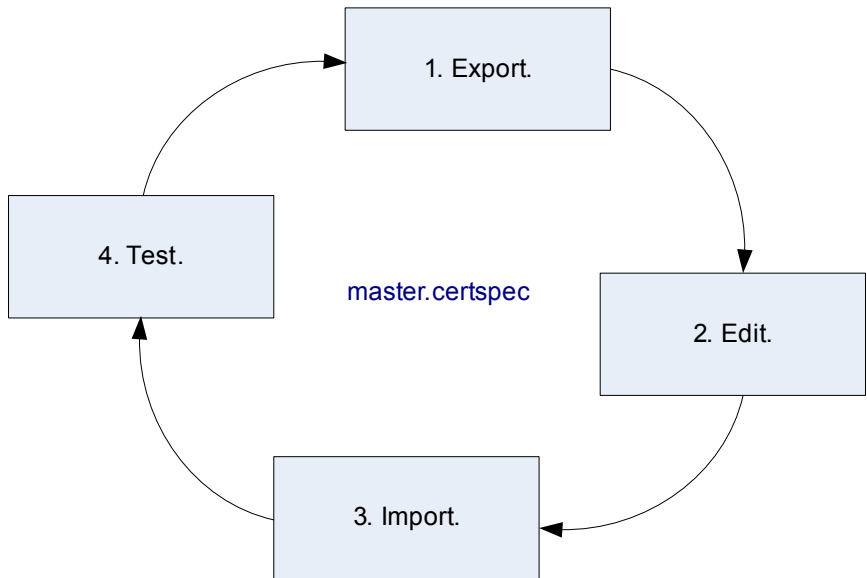
Before you can edit Security Manager certificate specifications, you must create and open the `master.certspec` file. You can then edit this file to customize certificates issued by Security Manager. After you have edited the `master.certspec` file, you process the file to write its contents to the Security Manager database.

The following diagram summarizes the steps you can take to customize the certificate specifications in the `initial.certspec` file:



After you initialize Security Manager, you must export the certificate specification file before you can make any changes to it.

The following diagram summarizes the steps you can take to customize the certificate specifications in the `master.certspec` file:



The following table indicates where you can find information about editing the `master.certspec` file to customize certificates issued by Security Manager.

For information about...	See...
Enterprise or Web certificates	<a href="#">"Customizing Enterprise and Web certificates" on page 544</a>
policy certificates	<a href="#">"Customizing policy certificates" on page 571</a>
cross-certificates	<a href="#">"Customizing cross-certificates" on page 580</a>
CA certificates	<a href="#">"Customizing CA certificates" on page 591</a>

Topics in this section:

- ["Creating the master.certspec file" on page 541](#)
- ["Opening the master.certspec file" on page 541](#)
- ["Navigating the master.certspec file" on page 542](#)
- ["Processing changes to the master.certspec file" on page 543](#)

## Creating the master.certspec file

Security Manager Administration creates the `master.certspec` file automatically when you export Security Manager certificate specifications from the Security Manager database. You export these certificate specifications using Security Manager Administration.

### To create the master.certspec file

- 1 In Security Manager Administration, click **File > Certificate Specifications > Export**.

The **Save As** dialog box appears.

- 2 Type a filename and location for the file and click **Save**.

By default, the file is named `master.certspec`.

A dialog box appears once the `master.certspec` file creates successfully.

- 3 Click **OK**.

You have now created the `master.certspec` file.

---

**Attention:** As a Security Officer, you are responsible for the `master.certspec` file. It is not recommended that you keep copies of this file. Keeping additional copies increases the risk of overwriting a newer version of the file with an older version.

---

## Opening the master.certspec file

Once you have created the `master.certspec` file, you can open it in any text editor. The following excerpt from the default `master.certspec` file shows the beginning of this file.

```
[Certificate Categories]
;*****
;* This section lists the categories to which certificate types may be *
;* assigned. Note that the order in which they are listed is not      *
;* important.                                                         *
;*                                                                     *
;* No changes to this section are required.                           *
;*****
1=enterprise
```

## Navigating the master.certspec file

The `master.certspec` file groups similar information into sections. You may modify the contents of several sections each time you edit the file.

To help you navigate the file, the beginning of each section is identified by its name enclosed in brackets. For example, the names of all the certificate categories are listed in the `[Certificate Categories]` section. When you need to work in a particular section, you can search for its section header using the search facility in the text editor.

---

**Attention:** Do not change the certificate categories listed in the `[Certificate Categories]` section. This section is reserved for future use, and any changes to it will cause errors when you attempt to import the certificate specifications.

---

The following excerpt from the default `master.certspec` file shows the major sections it contains.

```
[Certificate Categories]
...
[Certificate Types]
...
[Extension Definitions]
...
[Advanced Settings]
...
[Database Field Definitions]
...
[Attribute Definitions]
...
[Variables]
...
[Obsolete Certificate Types]
...
```

## Processing changes to the master.certspec file

After making your changes to the `master.certspec` file, you process the file. When you process the file, Security Manager Administration checks the file for errors. If Security Manager Administration finds errors, it writes descriptions of the errors to a log file that you specify (named `master.log` by default). Security Manager Administration never deletes the log file.

If Security Manager Administration does not find errors, it finishes processing the file, splitting lines longer than 72 characters into two or more lines. Additional lines are preceded by `_continue_`. For example:

```
web_default=web,Default,Default Web Certificates (for browsers and secure
_continue_=email)
```

Once Security Manager Administration finishes processing the file, Security Manager checks the file for errors. If Security Manager finds errors, descriptions of the errors display in a dialog box. If Security Manager does not find errors, it writes the contents of the file to the Security Manager database and signals Security Manager Administration to delete the `master.certspec` file. The `master.log` file contains a summary of the certificate types defined.

### To process the master.certspec file

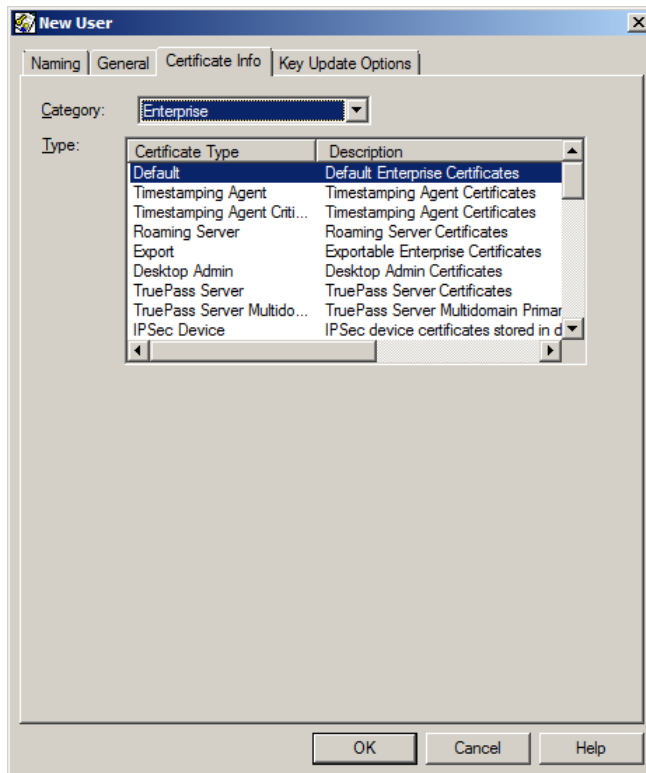
- 1** In Security Manager Administration, click **File > Certificate Specifications > Import**.  
The Open dialog box appears.
- 2** Select the `master.certspec` file in the **Open** dialog box and click **Open**.  
The **Save As** dialog box appears.
- 3** Specify a name and location (or accept the default) for a log file for any errors found by Security Manager Administration during the import, and click **Save**.  
A dialog box appears when the certificate specifications are imported.
- 4** Click **OK**.

You have now processed the `master.certspec` file.

# Customizing Enterprise and Web certificates

Certificates issued from the Enterprise and Web certificate categories provide useful information that identifies the certificate owner (the person the certificates are issued to). Certificates issued from these certificate categories are referred to collectively as user certificates. Security Manager issues user certificates when a user logs in for the first time, when a user's keys require updating, and when a user recovers lost keys.

You select the type of certificate that Security Manager issues to a particular user in the Certificate Info property page of the **New User** dialog box in Security Manager Administration. This example shows that the Default certificate type from the Enterprise category is assigned to this particular user. You can change the assigned certificate type by clicking another certificate type. For more information about the **New User** dialog box, see [“Modifying the user template file and user types” on page 447](#).



Within a certificate category, the difference among the types of certificates is the extra content that Security Manager adds to the certificate, in addition to the other X.509 content. For example, the previous example shows several certificate types for the Enterprise certificate category. All of these certificate types contain the same X.509 content. However, each certificate type is customized for a particular purpose.



For example, the Timestamp certificate type is customized to include extra content related to the Entrust Secure Transaction Platform Verification Server, while the Roaming Server certificate type is customized to include extra content related to the Roaming Server.

Topics in this section:

- [“Creating and modifying user certificate information” on page 545](#)
- [“Editing a certificate type description” on page 549](#)
- [“Editing certificate definitions” on page 551](#)
- [“Editing certificate extensions” on page 553](#)
- [“Defining a certificate extension” on page 557](#)
- [“Editing a variable” on page 565](#)
- [“Setting up default values” on page 567](#)
- [“Extended Key Usage \(EKU\) certificate extensions” on page 568](#)
- [“Certificate specification examples” on page 569](#)

## Creating and modifying user certificate information

If the certificate types listed in the **New User** dialog box do not meet your organization's requirements, you can create and modify certificate types. For example, you can create a certificate type in the Enterprise category that requires Security Manager to insert the job title of the certificate owner when issuing certificates for this certificate type.

---

**Note:** If the default certificate types meet your organization's requirements, you do not need to modify the defaults or create new ones.

---

You create and modify certificate types in the Enterprise and Web certificate categories using the `master.certspec` file. This section assumes that you know how to create, open, and process the `master.certspec` file. If you do not know how to perform these tasks, see [“Working with the master.certspec file” on page 539](#). In the `master.certspec` file, you can

- create new Enterprise and Web certificate types.  
After you create a certificate type and process the `master.certspec` file, the **New User** dialog box lists the new certificate type the next time you open it.
- modify the default Enterprise and Web certificate types.  
After you modify a certificate type and process the `master.certspec` file, Security Manager begins issuing certificates according to your changes.

You create or modify a certificate type using slightly different methods depending on whether it is a V1 or V2 certificate. (See [“What are V1 and V2 certificates?” on page 546.](#))

Topics in this section:

- [“What are V1 and V2 certificates?” on page 546](#)
- [“What is a certificate type description?” on page 546](#)
- [“What is a certificate extension?” on page 547](#)
- [“Using V1 certificate type descriptions and certificate extensions” on page 548](#)
- [“Using V2 certificate definitions and certificate extensions” on page 548](#)

## What are V1 and V2 certificates?

V1 and V2 certificates are those associated with V1-key-pair and V2-key-pair users, respectively. (For information on V1-key-pair and V2-key-pair users, see [“Configuring users’ key pairs” on page 107.](#)) Both types of certificates are specified in the `master.certspec` file in a similar way, but there are some differences between the two types.

Both V1 and V2 certificates are specified with

- a certificate type description
- certificate definitions
- certificate extensions

The main differences between the two types of certificates are

- V1 certificates have implied certificate definitions, whereas V2 certificates have explicit certificate definitions
- V2 certificates have a policy certificate associated with each certificate definition

---

**Note:** It is possible to use V1 certificates for V2 users, but to simplify the explanation, this guide describes them separately. It is also possible to use V2 certificate types for V1 users (as long as they have Encryption and Verification certificate definitions), but this practice is not recommended.

---

## What is a certificate type description?

The `master.certspec` file lists a certificate type description for each certificate type in the Enterprise and Web certificate categories. A certificate type description

- uniquely identifies the certificate type in the `master.certspec` file

- specifies the category that the certificate type belongs to
- provides the name and description for this certificate type when it is displayed in the **New User** dialog box

## What is a certificate definition?

A certificate definition describes a single key pair for a user or group of users.

Each V1 certificate type has one or two implicit certificate definitions—Encryption, Verification, or both. These certificate definitions are not explicitly listed in the `master.certspec` file, but each certificate extension is associated with either the Encryption or Verification certificate definition, or with both.

Each Enterprise V2 certificate type can have up to four associated certificate definitions, one for each of the user's key pairs. The V2 certificate definitions are listed explicitly in the `master.certspec` file.

Entrust provides a number of default V2 certificate types, each with one to four associated certificate definitions. For example, the Encrypted File System (EFS) User certificate type has three associated certificate definitions:

- Encryption
- Verification
- EFS encryption.

For basic information about V2 key-pair models, see [“Configuring users’ key pairs” on page 107](#).

The `master.certspec` file has an area for each default V2 certificate type, which contains a comment area that explains the certificate type, a section for the associated certificate definitions, and a section for each extension associated with the certificate definitions. Each certificate definitions section is headed by a section heading containing the name of the certificate type (for example, `[ent_efs Certificate Definitions]`).

In most cases, the default values in the certificate definitions are defined appropriately for their type of user. But in some cases, you may need to edit the certificate definitions or create a new certificate type. For example, if you have a user who needs the functionality from two different certificate types, you can define a new certificate type with the appropriate certificate definitions for each function.

## What is a certificate extension?

The `master.certspec` file may contain certificate extensions for each certificate type in the Enterprise and Web certificate categories. A certificate extension defines one value that Security Manager inserts in certificates issued for this certificate type. The value provided by an extension can be of a fixed or variable nature.

Certificate extensions defined in the `master.certspec` file define the extra content for certificates. These certificate extensions do not define the X.509 content that Security Manager inserts by default. However, you can use the `master.certspec` file to exclude these default certificate extensions.

Certificate extensions are used slightly differently depending on whether you are working with a V1 or V2 certificate. In a V1 certificate, the extensions are associated with either of the two implicit certificate definitions, Encryption or Verification, or with both. (See [“Using V1 certificate type descriptions and certificate extensions” on page 548.](#)) In a V2 certificate, the extensions are explicitly associated with each certificate definition. (See [“Using V2 certificate definitions and certificate extensions” on page 548.](#))

---

**Note:** A certificate type may contain database field definitions in addition to certificate extensions. Database fields are used to add content to the Security Manager database instead of certificates (see [“Customizing database fields” on page 601](#)). If a certificate type is created solely for the purpose of using database fields, the `master.certspec` file does not need to contain certificate extensions for that certificate type.

---

## Using V1 certificate type descriptions and certificate extensions

Understanding how certificate type descriptions and certificate extensions work together is best explained by example. Suppose that you want Enterprise users who work in branch offices to receive certificates that include their location (for example, Boston, Chicago, and so on).

To do this, you create two new certificate types in the Enterprise certificate category—one certificate type for users in the Boston office and one certificate type for users in the Chicago office. After creating the certificate types, you assign the appropriate certificate type in the **New User** dialog box to users in each branch office.

To create the certificate type for this example, you edit the `master.certspec` file to add the:

- certificate type description (see [“Editing a certificate type description” on page 549](#))
- certificate extension that specifies the name of the branch office to the implied Common certificate definition (see [“Editing certificate extensions” on page 553](#))

## Using V2 certificate definitions and certificate extensions

When you are working with V2 key pairs, explicit certificate definitions and extensions provide a method of defining the characteristics of a key pair. For example,

assume that you have two different groups of EFS users, one group that uses Windows Smart Card Logon, and one that does not.

To define the certificates for the two different groups, you use the default EFS User certificate type for the group that does not use Smart Card Logon, and create a new certificate type, such as EFS SCL User, for the group that does. You define the EFS SCL User certificate type to include the standard EFS User certificate definitions, and add an extension that specifies the Smart Card Logon extended key usage to one of the certificate definitions.

To create the certificate type for this example, you edit the `master.certspec` file to add the

- certificate type description (see [“Editing a certificate type description” on page 549](#))
- certificate definitions (see [“Editing certificate definitions” on page 551](#))
- certificate extension that specifies Smart Card Logon extended key usage (see [“Editing certificate extensions” on page 553](#))

---

**Note:** You cannot rename a certificate definition or remove one from a certificate type.

---

## Editing a certificate type description

Each certificate type for the Enterprise and Web certificate categories is represented by a certificate type description in the `master.certspec` file (see [“What is a certificate type description?” on page 546](#)).

In the `master.certspec` file, certificate type descriptions for all certificate types are located in the [Certificate Types] section. This example highlights the descriptions for the default certificate types in the Enterprise and Web certificate categories:

```
[Certificate Categories]
```

```
...
```

```
[Certificate Types]
```

```
...
```

```
ent_default=enterprise,Default,Default Enterprise Certificates
```

```
web_default=web,Default,Default Web Certificates (for browsers and secure email)
```

```
xcert_default=xcert,Default,Default Cross-Certificates
```

```
polcert_cliset=policycert,Client Settings,Client Setting Certificates
```

```
cacert_default=cacert,Default,Default Self Signed CA Certificates
```

```
cacert_link=cacert,Link,Self Issued Link CA Certificates
```

**Attention:** Once you add a certificate type, you cannot delete it. If your certificate type creates V2 key pairs, you also cannot make it obsolete. You can only make certificates with V1 key pairs obsolete ([“Making a certificate type obsolete” on page 599](#)).

**To add a certificate type description**

- 1** In the `master.certspec` file, locate the `[Certificate Types]` section header.
- 2** On a new line in the `[Certificate Types]` section, type the description for the certificate type.

You can add a new line anywhere in the `[Certificate Types]` section. Typically, however, you type descriptions for new certificate types after the default ones.

The following entry specifies a description for a certificate type. (Table 50 describes the parameters used in this entry.)

Type ID=Category ID,Type Name,Type Description

**Table 50:** Parameters in descriptions for user certificate types

Parameter	Description
Type ID	<p>Provides a unique representation for the certificate type in the <code>master.certspec</code> file (for example, <code>ent_default</code>).</p> <p>The characters may be either uppercase or lowercase. Security Manager converts all characters to lowercase when processing the <code>master.certspec</code> file.</p> <p>There is no maximum length for the Type ID, although the first 20 characters must be unique. If the Type ID exceeds 20 characters, it is truncated to 20 characters when Security Manager processes the <code>master.certspec</code> file.</p> <p>It is a good idea to identify the certificate category in the Type ID. For example, you can prefix the Type ID for Enterprise certificates with <code>ent_</code>.</p>
Category ID	<p>Identifies the user certificate category that this certificate type belongs to (that is, Enterprise and Web).</p>
Type Name	<p>Defines the name for this certificate type that appears in the list of certificate types in the New User dialog box.</p> <p>Note that the name cannot include commas or semicolons. It is limited to 20 characters.</p>
Type Description	<p>Defines the description for this certificate type that appears in the list of certificate types in the <b>New User</b> dialog box.</p> <p><b>Note:</b> The description cannot include commas or semicolons.</p>

The following example shows the values for the parameters in the description for the default Enterprise certificate type:

```
ent_default=enterprise,Default,Default Enterprise Certificates
```

Where:

- `ent_default` is the Type ID.
- `enterprise` is the Category ID.
- `Default` is the Type Name.
- `Default Enterprise Certificates` is the Type Description.

---

**Note:** Security Manager uses the default Enterprise certificate type for first-time initialization. Changes to this certificate type affect system dependencies throughout Security Manager.

---

- 3 Add a comment that describes the purpose of the certificate type when you add a description.

You are not required to add a comment, but you may find them useful. Comments help you remember the purpose of the new certificate type the next time you edit the file. To include a comment, type a semicolon followed by your comment. When Security Manager processes the file, it ignores everything between the semicolon and the end of the line.

You have now added a certificate type description.

## Editing certificate definitions

V1 certificate definitions are implicit. You cannot change them explicitly. You can only change them through the creation or editing of certificate extensions.

V2 certificate definitions are explicitly defined. Each V2 certificate type is represented by up to four certificate definitions in the `master.certspec` file (see [“What is a certificate definition?” on page 547](#)).

The list of the certificate definitions for a certificate type is provided under the section heading [`<Type ID> Certificate Definitions`]. This section is immediately followed by the extensions section for each certificate definition defined in the section. This example highlights the [`ent_nonrepudiation Certificate Definitions`] section.

```
...
[ent_nonrepudiation Certificate Definitions]
...
1=Encryption
2=Verification
```

```
3=NonRepudiation
```

```
[ent_nonrepudiation Encryption Extensions]  
keyusage=2.5.29.15,n,m,BitString,001
```

```
[ent_nonrepudiation Verification Extensions]  
keyusage=2.5.29.15,n,m,BitString,1
```

```
[ent_nonrepudiation NonRepudiation Extensions]  
keyusage=2.5.29.15,n,m,BitString,01  
...
```

---

**Attention:** Once you add a certificate definition, you cannot rename or delete it. Should you make the certificate type obsolete later by replacing it with a new certificate type, you must copy the certificate definitions from the obsolete certificate type to the replacement.

---

## To add a Certificate Definitions section

- 1 In the `master.certspec` file, locate the [`<Type ID>` Certificate Definitions] section header of the certificate definitions most similar to the one you want to create.

You can use this Certificate Definitions section as an example to follow in creating the new one.

- 2 On a new line in the file, type the section header in the form

```
[<Type ID> Certificate Definitions]
```

where `<Type ID>` represents the Type ID as defined in the [Certificate Types] section for the certificate type (for example, `ent_efs`). See [Table 50 on page 550](#) for details of the Type ID parameter.

Section headers must appear in square brackets and are case-sensitive.

- 3 On the next line, type the first certificate definition in the form

```
<n>=<Certificate Definition Name>
```

Where:

- `<n>` is the number of the certificate definition.
- `<Certificate Definition Name>` represents the certificate definition.

For example:

```
1=Encryption
```



- 4 Repeat [Step 3](#) for each certificate definition to be associated with the certificate type.
- 5 Add a comment that describes the purpose of the certificate type and definitions. You are not required to add a comment, but you may find it useful. Comments help you remember the purpose of the new certificate definitions the next time you edit the file. To include a comment, type a semicolon followed by your comment. When Security Manager processes the file, it ignores everything between the semicolon and the end of the line. Typically, a comment section that describes the key pairs associated with the certificate type precedes the Certificate Definitions section.

You have now added a Certificate Definitions section.

## Editing certificate extensions

Each certificate type for the Enterprise and Web certificate categories may contain certificate extensions in the `master.certspec` file (see [“What is a certificate extension?” on page 547](#)). Certificate extensions are defined differently for V1 and V2 certificate types. (See [“What are V1 and V2 certificates?” on page 546](#).) Although the format of the extension is the same for both types of certificates, their location in the `master.certspec` file and their section headers are different.

For more information about editing V1 certificate extensions, see

- [“Extensions for V1 certificate types” on page 553](#)
- [“Associating a certificate extension with a V1 certificate type” on page 554](#)
- [“To define a subsection header for a V1 certificate extension” on page 555](#)
- [“Defining a certificate extension” on page 557](#)

For more information about editing V2 certificate extensions, see

- [“Extensions for V2 certificate types” on page 555](#)
- [“Associating a certificate extension with a V2 certificate type” on page 556](#)
- [“To define a subsection header for a V2 certificate extension” on page 556](#)
- [“Defining a certificate extension” on page 557](#)

## Extensions for V1 certificate types

For each V1 certificate extension, you can specify whether you want its value included in encryption certificates, verification certificates, or both encryption and verification certificates for a certificate type. For example, suppose that you want Enterprise certificates to indicate whether the certificate owner is authorized to make purchases over \$500.00 on behalf of the company. When a user submits a purchase request, the person who receives the request can check the user's certificate before approving the purchase. This type of content is useful in verification certificates that

are used to verify a user's identity. There is no need for encryption certificates to include this content because they are not used to verify a user's identity.

In the `master.certspec` file, certificate extensions for V1 certificate types are located in the [Extension Definitions] section. The following example shows a certificate extension that defines extra content for verification certificates issued for the certificate type identified as `ent_timestamp` in the [Certificate Types] section.

```
[Extension Definitions]
...
[ent_timestamp Verification Extensions]
;-----
;- Timestamp Certificate Type                                -
;-                                                         -
;- This is a special certificate type that is to be used ONLY for -
;- creating certificates used to secure Timestamping Agents.      -
;- Timestamping Agents' verification certificates include the extended -
;- key usage extension which contains the id-kp-timeStamping usage -
;- identifier.                                                    -
;-----
extKeyUsage=2.5.29.37,n,m,SeqOfObjectIdentifier,1.3.6.1.5.5.7.3.8
...
```

## Associating a certificate extension with a V1 certificate type

Within the [Extension Definitions] section, certificate extensions are organized according to their certificate type, and whether the value provided by the extension is to be included in encryption, verification, or both encryption and verification certificates for that certificate type. The organization is identified using subsection headers. For example, certificate extensions for

- verification certificates issued for the `ent_default` certificate type are defined in the [ent\_default Verification Extensions] section
- encryption certificates issued for the `ent_default` certificate type are defined in the [ent\_default Encryption Extensions] section
- verification and encryption certificates issued for the `ent_default` certificate type are defined in the [ent\_default Common Extensions] section

---

**Note:** You cannot define certificate extensions defined as Common extensions to Encryption or Verification extensions.

---

## To define a subsection header for a V1 certificate extension

- 1 In the `master.certspec` file, locate the [Extension Definitions] section header.
- 2 On a new line in this section, type the subsection header.

Subsection headers for certificate extensions use the following formats:

```
[<Type ID> Verification Extensions]
```

```
[<Type ID> Encryption Extensions]
```

```
[<Type ID> Common Extensions]
```

where `<Type ID>` represents the Type ID as defined in the [Certificate Types] section for the certificate type (for example, `ent_default`). See [Table 50 on page 550](#).

Subsection headers must appear in square brackets and are case-sensitive.

---

**Note:** You only define subsection headers for the kinds of certificate extensions you define. For example, if you are not going to define certificate extensions for encryption certificates for the `ent_corporate` certificate type, you do not need to create the `[ent_corporate Encryption Extensions]` subsection header.

---

Once you have created a subsection within the [Extension Definitions] section, you can define its certificate extensions. See [“Defining a certificate extension” on page 557](#).

## Extensions for V2 certificate types

For V2 certificates, you can specify an extension for any of the certificate definitions associated with the certificate type. Typically, the extensions for a V2 certificate type follow the associated Certificate Definitions section of the `master.certspec` file. This example shows the extensions for the default EFS certificate type.

```
...
[ent_efs Certificate Definitions]
1=Encryption
2=Verification
3=EFS

[ent_efs Encryption Extensions]
keyusage=2.5.29.15,n,m,BitString,001

[ent_efs Verification Extensions]
keyusage=2.5.29.15,n,m,BitString,1
```

```
[ent_efs EFS Extensions]
keyusage=2.5.29.15,n,m,BitString,001
extKeyUsage=2.5.29.37,n,m,SeqOfObjectIdentifier,1.3.6.1.4.1.311.10.3.4
...
```

## Associating a certificate extension with a V2 certificate type

For V2 certificate types, certificate extensions are associated with certificate definitions. Typically, certificate extensions are organized in the same order as their associated certificate definitions. For example, in the EFS default certificate type shown in the previous example

- the encryption extensions are defined in the `[ent_efs Encryption Extensions]` section
- the verification extensions are defined in the `[ent_efs Verification Extensions]` section
- the EFS extensions are defined in the `[ent_efs EFS Extensions]` section

The Extensions section headers identify both the certificate type and the certificate definition associated with the extension.

### To define a subsection header for a V2 certificate extension

- 1 In the `master.certspec` file, locate the `[<Type ID> Certificate Definitions]` section header for the certificate type you want.
- 2 On a new line in this section, type the section header.

Section headers for certificate extensions use the following format:

```
[<Type ID> <Certificate Definition Name> Extensions]
```

where

- `<Type ID>` represents the Type ID as defined in the `[Certificate Types]` section for the certificate type (see [Table 50 on page 550](#) for details of the Type ID parameter),
- `<Certificate Definition Name>` is the name assigned to the appropriate certificate definition. For example:

```
[ent_efs EFS Extensions]
```

Section headers must appear in square brackets and are case-sensitive.

Once you create the Extensions section header, you can define its certificate extensions. See [“Defining a certificate extension” on page 557](#).

# Defining a certificate extension

You define a certificate extension in an Extensions section of the `master.certspec` file by specifying information about the value provided by the extension, and the value itself.

The value provided by a certificate extension can be of a fixed or variable nature. For example, one certificate extension may provide your organization's phone number. The phone number is a fixed value (that is, the value is the same in every certificate). A different certificate extension may provide the name of the department that the certificate owner belongs to. The department name is a variable value (that is, it is provided by the Entrust PKI administrator on a per-user basis when adding a user).

To define a certificate extension, type it in the appropriate extension section of the `master.certspec` file. Certificate extensions use the following syntax, which is described in the following topics:

Extn Name=Extn OID,Extn Crit,Extn Opt,Extn Type,Extn Value

**Note:** Security Manager uses the information you provide to encode the extension as an X.509 certificate extension. You are not coding an X.509 extension yourself.

For example, the following defines a certificate extension named `branchStreet`:

Extn Opt

Extn Crit

Extn Name

Extn OID

Extn Type

Extn Value

branchStreet=2.16.840.1.113730.1.13,n,o,IA5String,"750 Heron Road"

**Note:** The value 2.16.840.1.113730.1.13 is a sample OID that is used for illustration purposes only.

Topics in this section:

- ["Extn Name" on page 558](#)
- ["Extn OID" on page 558](#)
- ["Extn Crit" on page 559](#)
- ["Extn Opt" on page 560](#)
- ["Extn Type" on page 561](#)
- ["Extn Value" on page 563](#)

## Extn Name

The value of this parameter provides a unique representation for the certificate extension in the `master.certspec` file. For example, the Extn Name for the following certificate extension is `keyusage`:

```
keyusage=2.5.29.15,n,m,BitString,101
```

## Extn OID

The value of this parameter represents the object identifier (OID) of the certificate extension, in dotted decimal form. For example, the value `2.5.29.15` in the following certificate extension specifies the Extn OID:

```
keyusage=2.5.29.15,n,m,BitString,101
```

All extensions that you define must be registered and assigned object identifiers. If the extension is not registered, you must first obtain an OID for that extension by registering it. You obtain OIDs for extensions from your registration authority (for example, GTIS or ANSI).

If you use an OID in more than one certificate extension, you should use the same Extn Name for each of the certificate extensions.

Security Manager recognizes some Extn OIDs, some of which you can define in the `master.certspec` file and some of which you cannot. Table 51 lists the Extn OIDs and their corresponding Extn Names that Security Manager recognizes, but that you cannot define in the `master.certspec` file.

**Table 51:** Recognized Extn OIDs that you cannot use

OID	Extn Name
2.5.29.35	authorityKeyIdentifier
1.2.840.113533.7.65.0	entrustVersInfo
2.5.29.14	subjectKeyIdentifier
2.5.29.16	privateKeyUsagePeriod

Table 52 lists the Extn OIDs and Names that Security Manager recognizes, and that you can define in the `master.certspec` file. You define them in the `master.certspec` file to change their default criticality parameter (see [“Extn Crit” on page 559](#)) or value parameter (see [“Extn Value” on page 563](#)).

---

**Note:** For more information on the syntax for extension definitions, refer to RFC 3280, *Internet X.509 Public Key Infrastructure: Certificate and Certificate Revocation List (CRL) Profile*.

---

**Table 52:** Recognized Extn OIDs that you can use

OID	Extn Name	Define Value?	Define Criticality?	Comments
2.5.29.17		no	yes	Security Manager looks at the criticality only. The type and value are ignored. To set the criticality, see <a href="#">“Setting the criticality of the subjectAltName extension” on page 273</a> .
2.5.29.31	crlDistributionPoints	no	yes	Security Manager looks at the criticality only. The type and value are ignored.
2.5.29.19	basicConstraints	yes	yes	You cannot set pathLenConstraint for user certificates.  You must set the CA bit for the cross-certificate category.
2.5.29.32	certificatePolicies	yes	yes	If a definition of this extension is included in the certificate specification file, its value overrides the Encryption OIDs or Verification OIDs set under <b>Security Policy</b> in Security Manager Administration, or both.
2.5.29.15	keyUsage	yes	yes	The extension must be consistent with the section in which it is defined.  You cannot use this extension in a [ <code>&lt;Type ID&gt;</code> Common Extensions] section.

## Extn Crit

The value of this parameter determines whether the value provided by the certificate extension is critical. The value of this parameter is `c` (critical) or `n` (non-critical). For example, the value `n` in the following certificate extension indicates that the value provided by the extension is non-critical:

keyusage=2.5.29.15,n,m,BitString,101

For example, an extension may indicate that the user of a certificate can sign for purchases over \$1000. If the user signs for something and the application cannot interpret the \$1000 rule defined by the extension, the application then checks whether the extension is marked critical. If the extension is marked critical, the application rejects the certificate. When a certificate is rejected, the user is not able to sign for the purchase.

According to Recommendation X.509, when an application receives a certificate that includes an extension that the application does not know how to interpret, the application checks whether the extension is critical. If the application cannot interpret the extension, it must reject the certificate when the extension is marked as critical.

---

**Note:** Exercise caution when you mark an extension critical. Some applications cannot interpret a custom extension and the certificate is rejected. The solution is to build an application that can interpret and handle the custom extensions.

---

## Extn Opt

The value provided by this parameter determines whether the value provided by the certificate extension is mandatory. The value of this parameter is `o` (optional) or `m` (mandatory).

This parameter is important for certificate extensions that provide variable values (that is, values provided by the Entrust PKI administrator when enabling users). When Security Manager tries to issue a certificate that includes a variable certificate extension and the value for that certificate extension is missing (for example, the Entrust PKI administrator did not provide a value when enabling the user), Security Manager checks whether the extension is mandatory. Security Manager issues the certificate only if the certificate extension is optional. Security Manager does not issue a certificate if the certificate extension is mandatory.

Be careful about how you set the optionality for new certificate extensions with variables. If the extensions belong to types of certificates that are already issued to users, Security Manager tries to include the extension the next time that the certificates for these users are issued (for example, when the certificates are updated or when the users are set up for key recovery). If you set the extension as mandatory, Security Manager cannot issue the certificate. The value of the extension is not provided by the administrator for the existing users. You must re-enable each user and provide values for the new mandatory extension.

---

**Note:** Bulk commands provide a convenient method of performing operations for multiple users. For information about bulk commands, see [“Performing bulk operations” on page 275](#).

---



## Extn Type

The value provided by this parameter determines the data type of the value provided by the certificate extension. For example, the data format for the following certificate extension is `BitString`:

```
keyusage=2.5.29.15,n,m,BitString,101
```

Table 53 lists and describes the values that you can specify for the `Extn Type` parameter.

**Table 53:** Description of data formats

Use this data format... (single value)(ASN.1 sequence)		when the value provided by the extension is...
BitString	SeqOfBitString	a string of bits listed in binary notation, with the least significant bit first (for example, 1, 10100", 0110)
BMPString	SeqOfBMPString	a string of characters that are allowed for the corresponding ASN.1 type
Boolean	SeqOfBoolean	a value of 1 for TRUE or 0 for FALSE
DER	DER	in its ASN.1 DER-encoded representation (see <a href="#">"Extn Value" on page 563</a> )
GeneralizedTime	SeqOfGeneralizedTime	a string containing a "coordinated universal time" or Greenwich Mean Time (GMT) value as specified for the corresponding ASN.1 type (for example, 200409051200Z)
IA5String	SeqOfIA5String	(Same as BMPString.)
Integer	SeqOfInteger	an integer with a zero, positive, or negative value of any magnitude (for example, 0, 58, -5)
NumericString	SeqOfNumericString	(Same as BMPString.)
ObjectIdentifier	SeqOfObjectIdentifier	an object identifier, in dotted decimal form (for example, 2.5.29.15)
OctetString	SeqOfOctetString	a string of byte values listed in hexadecimal notation with two characters per octet and leading zeroes preserved (for example, 1A2B, 0C, 090B0D0F11). No spaces are allowed between octets.
PrintableString	SeqOfPrintableString	(Same as BMPString.)
Real	SeqOfReal	a real number with a zero, positive, or negative value of any magnitude (for example, 0, "1.813", -163.875, 1.345e-4)

**Table 53:** Description of data formats (continued)

Use this data format... (single value)(ASN.1 sequence)		when the value provided by the extension is...
UTCTime	SeqOfUTCTime	a string containing a “coordinated universal time” or Greenwich Mean Time (GMT) value as specified for the corresponding ASN.1 type (for example, 200809051200Z)
UTF8String	SeqOfUTF8String	(Same as BMPString.)

If the certificate extension provides a value of a variable nature, you define the variable (see [“Editing a variable” on page 565](#)). The data format that you specify for the Extn Type parameter determines the data format that you can specify for the variable when you define it. [Table 54 on page 562](#) lists the data formats that you can specify for the Extn Type and the corresponding variable types.

**Table 54:** Data formats for extension types and their variable types

Data type for certificate extension	Data type for variable
BitString	BitString, Boolean
BMPString	TextString
Boolean	Boolean
DER	OctetString
GeneralizedTime	DateTime
IA5String	TextString
Integer	Integer
NumericString	TextString
ObjectIdentifier	ObjectIdentifier
OctetString	OctetString
PrintableString	TextString
Real	Real
SeqOfBitString	BitStringList, BooleanList
SeqOfBMPString	TextStringList
SeqOfBoolean	BooleanList
SeqOfGeneralizedTime	DateTimeList

**Table 54:** Data formats for extension types and their variable types

Data type for certificate extension	Data type for variable
SeqOfIA5String	TextStringList
SeqOfInteger	IntegerList
SeqOfNumericString	TextStringList
SeqOfObjectIdentifier	ObjectIdentifierList
SeqOfOctetString	OctetStringList
SeqOfPrintableString	TextStringList
SeqOfReal	RealList
SeqOfUTCTime	DateTimeList
SeqOfUTF8String	TextStringList
UTCTime	DateTime
UTF8String	TextString

## Extn Value

This parameter identifies the value that Security Manager is to include in certificates. For example, the value for the following certificate extension is 101:

```
keyusage=2.5.29.15,n,m,BitString,101
```

Extensions or attributes with sequence types can also specify multiple constant values. Space-separate each value. For example, you could specify the `extendedKeyUsage` extension as:

```
extKeyUsage=2.5.29.37,n,m,SeqOfObjectIdentifier,1.2.840.113533.7.74.1
1.2.840.113533.7.74.2
```

Security Manager adds the value in its ASN.1 DER-encoded representation to the specified certificate extensions just before issuing the certificate. The ASN.1 recommendation for data structures is defined in *Recommendation X.680: Abstract Syntax Notation One (ASN.1)* and is maintained by the ITU-T. To represent ASN.1 structures in a machine-readable form, Security Manager encodes them using the Distinguished Encoding Rules (DER). These rules are just one of the methods defined in *Recommendation X.690: ASN.1 Encoding Rules* for encoding ASN.1 data structures. The X.690 recommendation is also maintained by the ITU-T.

For all of the formats listed in [Table 53 on page 561](#) except the DER format, Security Manager can convert the value provided by the Extn Value parameter to its ASN.1 DER-encoded format before inserting the value in a certificate.

For a fixed value that Security Manager can convert to its ASN.1 DER-encoded format, you simply type the value. For example, for the following certificate

extension, Security Manager inserts the ASN.1 DER-encoded representation of the fixed value 101 in certificates:

```
keyusage=2.5.29.15,n,m,BitString,101
```

If the fixed value is one of the text string types, enclose it in quotation marks and do not include semicolons. For example:

```
branchStreet=2.16.840.1.113730.1.13,n,o,IA5String,"750 Heron Road  
Ottawa"
```

Within the quotation marks, you can use the following characters, as long as you use a backslash (\) to identify them as part of the value:

- \ ("\\")
- " ("\"")
- < ("\\<")
- > ("\\>")
- line break ("\\n")
- tab ("\\t")

For example, this certificate extension adds the value "Apples\\Oranges" to the certificate:

```
comment=2.16.840.1.113730.1.13,n,m,IA5String,"\\Apples\\Oranges\\"
```

---

**Note:** Text string data types include: NumericString, PrintableString, IA5String, BMPString, UTF8String, SeqOfNumericString, SeqOfPrintableString, SeqOfIA5String, SeqOfBMPString, and SeqOfUTF8String.

---

For a value that Security Manager cannot convert to its ASN.1 DER-encoded format (a value that cannot be represented by one of the data formats listed in [Table 53 on page 561](#) other than DER), you must provide the ASN.1 DER-encoded representation of the value. The representation must contain two characters per octet, without spaces between octets, and the representation must preserve leading zeroes.

---

**Note:** If you want to include a complex value in a certificate extension and you do not know how to convert the value to its ASN.1 DER-encoded representation, contact the ITU-T for *Recommendation X.690: ASN.1 Encoding Rules*.

---

For any variable value, type the variable name enclosed in angle brackets ("<>"). For example, the following certificate extension uses a variable named "branchstreet":

```
branchStreet=2.16.840.1.113730.1.13,n,o,IA5String,"<branchstreet>"
```

If you use a variable, you must define it in the [Variables] section of the master.certspec file (see ["Editing a variable" on page 565](#)).

# Editing a variable

Certificate extensions define a value (of a fixed or variable nature) that Security Manager is to insert in certificates. A certificate extension that provides a variable value specifies a variable name for its `Extn Value` parameter (see [“Extn Type” on page 561](#)). You must define the variable in the `[Variables]` section of the `master.certspec` file.

## To define a variable

- 1 Locate the `[Variables]` section in the `master.certspec` file.
- 2 On a new line in this section, define the variable using the syntax shown in the [Table 55 on page 565](#). (This table describes the parameters used in variable definitions.)

```
Var ID=Var Type,Var Label,Var Desc,[Range | OneOf],Value Spec
1,<Value Spec 2>,...
```

**Table 55:** Parameters for variable definitions

Parameter	Description
Var ID	<p>Names the variable. The name must be unique and must match the name used in the <code>Extn Value</code> parameter of the certificate extension as defined in the <code>[Extension Definitions]</code> section.</p> <p>The characters may be either uppercase or lowercase. Security Manager converts all characters to lowercase when processing the <code>master.certspec</code> file.</p> <p>There is no maximum length for the Var ID, although the first 20 characters must be unique. If the Var ID exceeds 20 characters, it is truncated to 20 characters when Security Manager processes the <code>master.certspec</code> file.</p>
Var Type	<p>Identifies the data type of the variable. If the value provided for the variable is a single value, the following data types are valid:</p> <ul style="list-style-type: none"><li>• Boolean</li><li>• Integer</li><li>• BitString</li><li>• OctetString</li><li>• Real</li><li>• ObjectIdentifier</li><li>• TextString</li><li>• DateTime</li></ul>

**Table 55:** Parameters for variable definitions (continued)

Parameter	Description
Var Type (continued)	<p>To specify a variable with multiple values, use one of the following types:</p> <ul style="list-style-type: none"> <li>• BooleanList</li> <li>• IntegerList</li> <li>• BitStringList</li> <li>• OctetStringList</li> <li>• RealList</li> <li>• ObjectIdentifierList</li> <li>• TextStringList</li> <li>• DateTimeList</li> </ul> <p>Note that the value you specify for the <code>Var Type</code> parameter must correspond to the value specified for the <code>Extn Type</code> parameter of the certificate extension (see “<a href="#">Extn Type</a>” on page 561).</p>
Var Label	Defines the prompt that appears in Security Manager Administration dialog boxes (for example, “User’s phone number:”).
Var Desc	Defines the help description that appears in Security Manager Administration dialog boxes (for example, the values that are valid for this variable).
Range or OneOf	Indicates whether the value provided by this parameter must lie between a minimum and maximum range, or whether the value must be one of a list of valid values. The parameter you choose determines the number of <code>Value Spec n</code> parameters you must specify.
Value Spec n	<p>For a range of values, you must provide the minimum and maximum values using two <code>Value Spec n</code> parameters. Use <code>Value Spec 1</code> for the minimum number or length and <code>Value Spec 2</code> for the maximum number or length.</p> <p>For the <code>OneOf</code> parameter, you specify each value that is selectable using one <code>Value Spec n</code> parameter.</p> <p>The <code>Value Spec n</code> parameter cannot include commas, semicolons, or spaces.</p>

For example, the following variable definition specifies that the Entrust PKI administrator must choose a value between 0 and 52 weeks:

```
password_lifetime=Integer,Password expires in (weeks):,Number of
weeks that passwords remain valid. (0 = never expire),Range,0,52
```

The following variable definition specifies that the Entrust PKI administrator must choose one of the four values specified (ALL, SHA-1, MD5, or SHA-256):

```
allowed_hashing_algs=TextStringList,Hashing
algorithms:,Algorithm(s) which may be used to hash a user's
data.,OneOf, "ALL", "SHA1", "MD5", "SHA256"
```

## Setting up default values

In Security Manager, you can set up a default value for a variable specified in an extension or attribute in the `master.certspec` or `initial.certspec` file. The default value is used to encode user or attribute certificates when no variable is available. This means that you can add a new extension or attribute to a certificate type without a new value being added to all of the users (or policies) using that certificate type.

You may want to set up default values for the variable definitions if you plan to use any of the following user policies:

- `permit_server_login`
- `enforce_identity_use`
- `http_proxy_setting`

You can specify the default value in the [Variables] section of the `master.certspec` or `initial.certspec` file as follows:

```
<Var ID>=<Var Type>,<Var Label>,<Var Description>,[<Default
Value>,<Range | OneOf>,<Value 1>,<Value 2>,...]
```

You can also specify the default value in the [Default Variable Values] section of the `entmgr.ini` file as follows:

```
<Var ID>=<Default Value>
```

If the [Default Variable Values] section does not appear in the `entmgr.ini` file, you must add it the first time you add a default value, as follows:

```
[Default Variable Values]
<Var ID>=<Default Value>
```

The [Default Variable Values] section can go anywhere in the `entmgr.ini` file.

---

**Note:** If `<Var Type>` is `TextString`, you must enclose the default value in quotation marks (" ") in the certificate specifications file, or attempting to import the file back into Security Manager will fail. If you set the default value in the `entmgr.ini` file, do not enclose the default value in quotation marks.

---

Once a default value is established, you can then add new extensions or attributes with new variables to a certificate from Security Manager Administration without having to modify certificates that already exist.

## To add a default value to a variable definition

- 1 To add a default value to a variable definition, you need to customize the certificate.

To customize certificates, you edit the `master.certspec` file. Before you can edit this file, you need to export it. In Security Manager Administration, click **File > Certificate Specifications > Export**.

- 2 Place the default value before the `Range` or `OneOf` field.

Note that the default is optional, and if the default value is not specified it implies that no default value is available.

In this example:

```
login_message=TextString,Message in Entrust Ready clients:,Message  
text to appear in Entrust Ready application., "Hello  
World",Range,0,80
```

the words "Hello World" are the default value.

- 3 Save your changes and import the revised `master.certspec` file.

In Security Manager Administration, click **File > Certificate Specifications > Import**.

When the `master.certspec` (or `initial.certspec`) file is imported, any default values are validated by Security Manager Administration against the variable definition. If the value is invalid, an error message displays.

You have now added a default value to a variable definition.

## Extended Key Usage (EKU) certificate extensions

Entrust certificates contain a public key and a value that defines how that key is used. The key usage certificate extension provides the value in a certificate that defines how to use that certificate's public key. The key usage certificate extension must specify a purpose defined by the X.509 recommendation. For example, one certificate may contain a value that indicates that the public key is used only to digitally sign information. Another certificate may contain a value that indicates that the public key is used to digitally sign information and to encrypt information.

Some Microsoft PKI-enabled applications may also require an extended key usage (EKU) certificate extension. The value in the EKU certificate extension identifies the purpose of the certificate's public key in a form that the Microsoft PKI-enabled application can use.

You can find requirements for EKU certificate extensions in the IETF RFC 3280, *Internet X.509 Public Key Infrastructure Certificate and CRL Profile* (as of January 1999).

For detailed information about EKU certificate extensions, see the *Security Manager Interoperating with Microsoft® PKI-enabled applications* document.



A list of common key usages is shown in Table 56.

**Table 56:** Common Key Usages

1.3.6.1.5.5.7.3.1	PKIX - Server Authentication	Indicates that you can use a certificate as an SSL server certificate.
1.3.6.1.5.5.7.3.2	PKIX - Client Authentication	Indicates that you can use a certificate as an SSL client certificate.
1.3.6.1.5.5.7.3.3	PKIX - Code Signing	Indicates that you can use a certificate for code signing.
1.3.6.1.5.5.7.3.4	PKIX - email	Indicates that you can use a certificate to protect email (signing, encryption, key agreement).
1.3.6.1.5.5.7.3.5	PKIX - IPSec End System	IPSEC End System Certificate.
1.3.6.1.5.5.7.3.6	PKIX - IPSec Tunnel	IPSEC Tunnel Certificate.
1.3.6.1.5.5.7.3.7	PKIX - IPSec User	IPSEC User Certificate.
1.3.6.1.5.5.7.3.8	PKIX - Timestamping	Indicates that you can use a certificate to bind the hash of an object to a time from a trusted time source.
1.3.6.1.4.1.311.10.3.1	MS - Certificate Trust List (CTL)	A list of trusted Certificate Authorities.
1.3.6.1.4.1.311.10.3.3	MS - Microsoft Server Gated Crypto (SGC)	An encryption improvement for old browsers to allow users to get strong cryptography.
1.3.6.1.4.1.311.10.3.4	MS - Microsoft Encrypted File System (EFS)	Indicates that the certificate allows the holder to use EFS to encrypt and decrypt data.
2.16.840.1.113730.4.1	Netscape Server Gated Crypto	Strong crypto export approved.
1.3.6.1.4.1.311.20.2.2	Smart Card Logon	Required to be present in certificates for Microsoft Smart Card Logon for Windows.

## Certificate specification examples

The following excerpt from an example V2 `master.certspec` file shows the additions to the `master.certspec` file to customize a certificate type for a sample EFS user who also requires the Smart Card Logon capability.

```
[Certificate Categories]
```

```
...
```

```

[Certificate Types]
...
ent_efs_scl=enterprise,EFS SCL User,EFS and Smart Card Logon User
...
[ent_efs_scl Certificate Definitions]
1=Encryption
2=Verification
3=EFS

[ent_efs_scl Encryption Extensions]
extKeyUsage=2.5.29.37,n,m,SeqOfObjectIdentifier,1.3.6.1.5.5.7.3.2
1.3.6.1.4.1.311.20.2.2
keyusage=2.5.29.15,n,m,BitString,001

[ent_efs_scl Verification Extensions]
extKeyUsage=2.5.29.37,n,m,SeqOfObjectIdentifier,1.3.6.1.5.5.7.3.2
1.3.6.1.4.1.311.20.2.2
keyusage=2.5.29.15,n,m,BitString,1

[ent_efs_scl EFS Extensions]
keyusage=2.5.29.15,n,m,BitString,001
extKeyUsage=2.5.29.37,n,m,SeqOfObjectIdentifier,1.3.6.1.4.1.311.10.3.4
...

```

# Customizing policy certificates

Certificates issued from the policy certificate category define user policies. User policies determine what users who belong to a particular Entrust role, or who are assigned a particular certificate definition, can and cannot do within Security Manager (see [“Administering user policies” on page 391](#)).

Security Manager issues policy certificates when you create or modify a user policy. Users are governed by the policy certificate associated with the user role they are assigned. For V2 users, there is also a policy certificate associated with each certificate definition assigned to their certificate type. (See [“What are V1 and V2 certificates?” on page 546](#).)

You create a user policy using the **New User Policy** dialog box in Security Manager Administration. In this dialog box, you choose the type of policy certificate that you want Security Manager to issue. Currently one certificate type, named Client Settings, is available for association with user roles. For V2 users, there is also another policy certificate type, named Cert. Defn. Settings, for association with V2 certificate definitions.

The certificate type you choose determines the attributes that appear in the lower portion of the dialog box. This example shows attributes that appear when you select the Client Settings certificate type.

**New User Policy**

Label:

Common name:

Add to:

Category:

Type:   
Client Settings (Client Setting Certificates)   
Cert. Defn. Settings (Policy Settings for Certificate Definitions)

Policy Attributes

Password expires in (weeks):

Password history:

Password length (characters):

Password needs non-alpha char.: ☐

After you provide values for the certificate attributes and click **OK**, Security Manager issues the policy certificate, including the attributes you have defined. Once certificate attributes are included in a policy certificate, they are referred to as user policies. For more information about policy values and defaults, see [“Client policy attributes reference” on page 407](#) and [“Certificate definition policy attributes reference” on page 432](#).

---

**Note:** Do not modify either of the default policy certificates (Client Settings or Cert. Defn. Settings) without instructions from Entrust.

---

Topics in this section:

- [“Creating and modifying a certificate type” on page 572](#)
- [“Editing a certificate type description” on page 573](#)
- [“Editing certificate attributes” on page 575](#)
- [“Defining a certificate attribute in a subsection” on page 576](#)
- [“Editing a variable for a certificate attribute” on page 578](#)
- [“Enforcing the use of custom policy certificates” on page 578](#)

## Creating and modifying a certificate type

If the certificate type listed in the **New User Policy** dialog box does not meet the requirements of your organization, you can add attributes to this certificate type. You cannot remove or modify the predefined attributes. You can also define new certificate types in the policy certificate category.

---

**Note:** Entrust does not intend for you to create certificate types in the policy certificate category for use with Entrust desktop applications. Entrust provides this feature to enable you to create policy certificates for your own purposes.

---

You create and modify certificate types in the policy certificate category using the `master.certspec` file. This section assumes that you know how to create, open, and process the `master.certspec` file. If you do not know how to perform these tasks, see [“Working with Security Manager certificate specifications” on page 531](#). In the `master.certspec` file, you can

- create new policy certificate types.

After you create a new certificate type and process the `master.certspec` file, the **New User Policy** dialog box lists the new certificate type the next time you open it.

- add attributes to the default policy certificate type.

After you modify the default policy certificate type and process the `master.certspec` file, Security Manager begins issuing certificates according to your changes.

You create or modify a certificate type using a certificate type description and one or more certificate attributes.

## What is a certificate type description?

The `master.certspec` file contains a certificate type description for each certificate type in the policy certificate category. A certificate type description

- uniquely identifies the certificate type in the `master.certspec` file
- specifies the certificate category that the certificate type belongs to
- specifies the name and description for this certificate type when it is displayed in the New User Policy dialog box

## What is a certificate attribute?

The `master.certspec` file contains one or more certificate attributes for each certificate type in the policy certificate category. A certificate attribute defines one value that Security Manager inserts in certificates issued for this certificate type. The value provided by an attribute can be of a fixed or variable nature.

## Editing a certificate type description

Each certificate type for the policy certificate category is represented by a certificate type description (see [“What is a certificate type description?” on page 573](#)).

In the `master.certspec` file, certificate type descriptions for all certificate types are located in the `[Certificate Types]` section. This example highlights the description for the default certificate type in the policy certificate category:

```
[Certificate Categories]
...
[Certificate Types]
...
; -----
; Default Certificate Types
;
; These type definitions must not be removed. Their <Type Name>
; and <Type Description> fields may be changed, however.
; -----
ent_default=enterprise,Default,Default Enterprise Certificates
```

```
web_default=web,Default,Default Web Certificates (for browsers and secure email)
xcert_default=xcert,Default,Default Cross-Certificates
polcert_cliset=policycert,Client Settings,Client Setting Certificates
[cacert_default=cacert,Default,Default Self Signed CA Certificates
cacert_link=cacert,Link,Self Issued Link CA Certificates
...
```

**To add a certificate type description**

- 1 In the master.certspec file, locate the [Certificate Types] section header.
- 2 On a new line in the [Certificate Types] section, type the description for the certificate type.

You can add a new line anywhere in the [Certificate Types] section. Typically, however, you type descriptions for new certificate types after the default ones.

The following entry specifies a description for a certificate type. (Table 57 on page 574 describes the parameters used in descriptions for policy certificate types, except for the Category ID, which is always policycert.)

Type ID=policycert,Type Name,Type Description

**Attention:** Do not modify the Type ID or Category ID parameters for the default certificate type. Changes to these parameters may cause serious problems with your user policies.

**Table 57:** Parameters in policy certificate type descriptions

Parameter	Description
Type ID	<p>Provides a unique representation for the certificate type in the master.certspec file (for example, polcert_cliset).</p> <p>The characters may be either uppercase or lowercase. Security Manager converts all characters to lowercase when processing the master.certspec file.</p> <p>There is no maximum length for the Type ID, although the first 20 characters must be unique. If the Type ID exceeds 20 characters, it is truncated to 20 characters when Security Manager processes the master.certspec file.</p> <p>It is a good idea to identify the certificate category in the Type ID. For example, you can prefix the Type ID for policy certificate types with polcert_.</p>
Type Name	<p>Defines the name for this certificate type that appears in the list of certificate types in the <b>New User Policy</b> dialog box.</p> <p>Note that the name cannot include commas or semicolons. It is limited to 20 characters.</p>

**Table 57:** Parameters in policy certificate type descriptions (continued)

Parameter	Description
Type Description	Defines the description for this certificate type that appears in the list of certificate types in the <b>New User Policies</b> dialog box.  Note that the description cannot include commas or semicolons.

For example, here are the values for the parameters that specify the description for the default policy certificate type:

Type ID	Category ID	Type Name	Type Description
polcert_cliset=polcycert,Client Settings,Client Setting Certificates			

- 3** Add a comment that describes the purpose of the certificate type when you add a description.
- You are not required to add a comment, but you may find them useful. Comments help you remember the purpose of the new information the next time you edit the file. To include a comment, type a semicolon, followed by your comment. When Security Manager processes the file, it ignores everything between the semicolon and the end of the line.

You have now added a certificate type description.

## Editing certificate attributes

Each certificate type for the policy certificate category contains certificate attributes (see [“What is a certificate attribute?” on page 573](#)) in the `master.certspec` file.

In the `master.certspec` file, certificate attributes for all policy certificate types are located in the `[Attribute Definitions]` section. Within the `[Attribute Definitions]` section, certificate attributes are organized according to their certificate type. The organization is identified using subsection headers. For example, certificate attributes for the `polcert_cliset` certificate type are defined in the `[polcert_cliset Attributes]` subsection of the `[Attribute Definitions]` section.

### To define a subsection header for a certificate attribute

- 1** In the `master.certspec` file, locate the `[Attribute Definitions]` section header.
- 2** On a new line in this section, type the subsection header.  
Subsection headers for certificate attributes use the following format:

[<Type ID> Attributes]

where <Type ID> represents the Type ID as defined in the [Certificate Types] section for the certificate type (for example, [polcert\_cliset Attributes]).

Subsection headers must appear in square brackets and are case-sensitive.

Once you have created a subsection within the [Attribute Definitions] section, you can define its certificate attributes.

## Defining a certificate attribute in a subsection

You define a certificate attribute within a subsection of the [Attribute Definitions] section of the master.certspec file by specifying information about the value provided by the attribute, and the value itself.

The value provided by a certificate attribute can be of a fixed or variable nature. For example, one certificate attribute for the default policy certificate type defines the lifetime of certificates. The lifetime is a variable value, provided by the Entrust PKI administrator on a per-policy basis when creating the user policy.

---

**Attention:** Do not modify the defined certificate attributes for the default policy certificate type. Changes may cause serious problems with your user policies.

---

To define a certificate attribute, type it in the appropriate subsection of the [Attribute Definitions] section. Certificate attributes use the following syntax, which is described in the following topics:

Attr Name=Attr OID,Attr Type,Attr Value

---

**Note:** Security Manager uses the information you provide to encode it as an X.509 certificate attribute. You are not coding an X.509 attribute yourself.

---

For example, the following defines a certificate attribute named policy\_cert\_lifetime.

Attr Name	Attr OID	Attr Type	Attr Value
policy_cert_lifetime	=1.2.840.113533.7.77.13	Integer	<policy_cert_lifetime>

---

**Note:** The value 1.2.840.113533.7.77.13 is an OID that is registered for this attribute.

---

Topics in this section:

- [“Attr Name” on page 577](#)



- [“Attr OID” on page 577](#)
- [“Attr Type” on page 577](#)
- [“Attr Value” on page 578](#)

## Attr Name

The value of this parameter provides a unique representation for the certificate attribute in the `master.certspec` file. For example, the `Attr Name` for the following certificate attribute is `allowed_hashing_algs`:

```
allowed_hashing_algs=1.2.840.113533.7.77.4,SeqOfUTF8String,"<allow
ed_hashing_algs>"
```

## Attr OID

The value of this parameter represents the object identifier (OID) of the certificate attribute, in dotted decimal form. For example, the value `2.840.113533.7.77.4` in the following certificate attribute specifies the OID:

```
allowed_hashing_algs=1.2.840.113533.7.77.4,SeqOfUTF8String,"<allow
ed_hashing_algs>"
```

You must register all attributes that you define and assign them object identifiers. If the attribute is not registered, you must first obtain an OID for that attribute by registering it. You obtain OIDs for attributes from your registration authority (for example, GTIS or ANSI).

If you use an OID in more than one certificate attribute, you should use the same `Attr Name` for the certificate attributes.

## Attr Type

The value of this parameter identifies the data format of the value provided by the certificate attribute. For example, the data format for the following certificate attribute is `SeqOfUTF8String`:

```
allowed_hashing_algs=1.2.840.113533.7.77.4,SeqOfUTF8String,"<allow
ed_hashing_algs>"
```

All the data formats that are valid for certificate extensions are also valid for certificate attributes. [Table 53 on page 561](#) lists and describes the values that you can specify for the `Extn Type` parameter. The same values are available for the `Attr Type` parameter. One additional data format is valid for certificate attributes. This data format is named `PasswordRules` and its values include a set of Entrust-defined password rules.

If the certificate attribute provides a value of a variable nature, you must define the variable (see [“Editing a variable” on page 565](#)). The data format that you specify for the `Attr Type` parameter determines the data format that you can specify for the

variable when you define it. [Table 54 on page 562](#) lists the data formats that you can specify for the `Extn Type` and the corresponding variable types. The same formats and variable types are available for `Attr Type`. One additional format is available for `Attr Type`. This data type is named `PasswordRules`, and its corresponding variable data types are `Integer` and `Boolean`.

## Attr Value

The value of this parameter identifies the value that Security Manager is to include in certificates for this attribute. For example, the value for the following certificate attribute is provided by a variable named `allowed_hashing_algs`:

```
allowed_hashing_algs=1.2.840.113533.7.77.4,SeqOfUTF8String,"<allowed_hashing_algs>"
```

Security Manager adds the value in its ASN.1 DER-encoded representation to certificates just before issuing them. The ASN.1 recommendation for data structures is defined in *Recommendation X.680: Abstract Syntax Notation One (ASN.1)* and is maintained by the ITU-T. To represent ASN.1 structures in a machine-readable form, Security Manager encodes them using the Distinguished Encoding Rules (DER). These rules are just one of the methods defined in *Recommendation X.690: ASN.1 Encoding Rules* for encoding ASN.1 data structures. The X.690 recommendation is also maintained by the ITU-T.

For all of the formats listed in Table 53 except the DER format, Security Manager can convert the value provided by the `Attr Value` parameter to its ASN.1 DER-encoded format before inserting the value in a certificate.

---

**Note:** If you want to include a complex value in a certificate extension and you do not know how to convert the value to its ASN.1 DER-encoded representation, contact the ITU-T for *Recommendation X.690: ASN.1 Encoding Rules*.

---

## Editing a variable for a certificate attribute

If you define a certificate attribute that names a variable (instead of a fixed value), you must define the variable. Variables are defined in the `[Variables]` section of the `master.certspec` file.

You define variables for certificate attributes in the same way that you define variables for certificate extensions. For information, see ["Editing a variable" on page 565](#).

## Enforcing the use of custom policy certificates

By default in Security Manager, when a user logs in, the client software first looks for the user's policy certificate in the directory. If the client cannot connect to the

directory (for example, if the user is working offline), the client then looks for the user's `.pch` file, which is a locally-stored, cached copy of the user's policy certificate. If this cannot be found, the user logs in with default user policy settings (see [“Client policy attributes reference” on page 407](#) for a description of each policy setting, including the default for each).

If you wish to enforce the use of custom policy certificates that your organization has established (you do not want users logging in using the default user policy settings), you can add an extension to user certificates that requires users to have a valid PCH file when they log in. This forces users to have a valid user policy from the directory or from a cached PCH file. To enforce the use of policy certificates, you must edit the `master.certspec` file.

### To enforce the use of policy certificates

- 1 Log in to Security Manager Administration.
- 2 Click **File > Certificate Specifications > Export**.

Save the file to a known location on disk.

- 3 Open the file in a text editor and add the following section:

```
[ent_default Verification Extensions]
;-----
; Require .PCH during login
;-----
require_policy_cert=2.16.840.1.114027.30.2,n,m,DER,0500
```

- 4 Click **File > Save** and close the file.
- 5 In Security Manager Administration, click **File > Certificate Specifications > Import** to import the updated `master.certspec` file.

Performing this procedure adds the extension to every verification certificate created with the Default certificate type. To add the extension to other certificate types, add the code again, but replace `ent_default` with the relevant identifier.

---

**Note:** This change only affects newly created verification certificates. If you add this extension to the `master.certspec` but do not update the users' verification certificates, they can continue to log in offline with no PCH file. If there is a requirement to have different types of users in the same CA (some of whom require the PCH file and some of whom do not), you must have two or more certificate types—at least one with an extension and one without.

---

# Customizing cross-certificates

Certificates issued from the cross-certificate category enable CAs to establish direct trust. Security Manager issues cross-certificates during the cross-certification process (see [“Cross-certifying with other CAs” on page 465](#)).

During the cross-certification process, you are prompted to choose a cross-certificate type. The cross-certificate type you choose determines the extra content that Security Manager adds to the certificate, in addition to the standard X.509 content.

The dialog box that prompts you to choose a certificate type depends on the cross-certification method you are using, and whether you are initiating or completing cross-certification. For any of the dialog boxes, you can change the certificate type that is used to generate a cross-certificate by clicking another certificate type.

Topics in this section:

- [“Creating and modifying a cross-certificate type” on page 580](#)
- [“Adding a certificate type description” on page 582](#)
- [“Editing certificate extensions” on page 584](#)
- [“Examples of cross-certificate types” on page 589](#)

## Creating and modifying a cross-certificate type

If the certificate types listed in the cross-certification dialog boxes do not meet the requirements of your organization, you can create and modify certificate types. For example, you can create a certificate type in the cross-certificate category that specifies that only a specific subgroup of users in the cross-certified CA are trusted.

---

**Note:** If the default certificate type meets the requirements of your organization, you do not need to modify it or create new certificate types.

---

You create and modify certificate types in the cross-certificate category using the `master.certspec` file. This section assumes that you know how to create, open, and process the `master.certspec` file. If you do not know how to perform these tasks, see [“Working with the master.certspec file” on page 539](#). In the `master.certspec` file, you can

- create new cross-certificate types.

After you create a certificate type and process the `master.certspec` file, the cross-certification dialog boxes list the new certificate type the next time they appear.

- modify the default cross-certificate types.

After you modify a certificate type and process the `master.certspec` file, Security Manager begins issuing certificates according to your changes.

You create or modify a certificate type using a certificate type description and one or more certificate extensions.

## What is a certificate type description?

The `master.certspec` file contains a certificate type description for each certificate type in the cross-certificate category. A certificate type description:

- uniquely identifies the certificate type in the `master.certspec` file
- specifies the category that the certificate type belongs to
- provides the name and description for this certificate type when it is displayed in the cross-certification dialog boxes

## What is a certificate extension?

The `master.certspec` file may contain certificate extensions for each certificate type in the cross-certificate category. A certificate extension defines one value that Security Manager inserts in certificates issued for this certificate type. The value provided by an extension for a cross-certificate type must be of a fixed nature.

Certificate extensions defined in the `master.certspec` file define the extra content for certificates. These certificate extensions do not define the X.509 content that Security Manager inserts by default. However, you can use the `master.certspec` file to exclude certificate extensions that add X.509 content.

## Using certificate type descriptions and certificate extensions

To create a new certificate type within the cross-certificate category, you edit the `master.certspec` file to add the

- certificate type description (see [“Adding a certificate type description” on page 582](#))
- certificate extension (see [“Editing certificate extensions” on page 584](#))

## Adding a certificate type description

Each certificate type for the cross-certificate category is represented by a certificate type description (see [“What is a certificate type description?” on page 581](#)) in the `master.certspec` file.

In the `master.certspec` file, certificate type descriptions for all certificate types are located in the `[Certificate Types]` section. This example highlights the description for the default certificate type in the cross-certificate category:

```
[Certificate Categories]
...
[Certificate Types]
...
; -----
; Default Certificate Types
;
; These type definitions must not be removed. Their <Type Name>
; and <Type Description> fields may be changed, however.
; -----
ent_default=enterprise,Default,Default Enterprise Certificates
web_default=web,Default,Default Web Certificates (for browsers and secure email)
xcert_default=xcert,Default,Default Cross-Certificates
polcert_cliset=policycert,Client Settings,Client Setting Certificates
cacert_default=cacert,Default,Default Self Signed CA Certificates
cacert_link=cacert,Link,Self Issued Link CA Certificates
...
```

### To add a certificate type description

- 1 In the `master.certspec` file, locate the `[Certificate Types]` section header.
- 2 On a new line in the `[Certificate Types]` section, type the description for the certificate type.

You can add a new line anywhere in the `[Certificate Types]` section. Typically, however, you type descriptions for new certificate types after the default ones.

The following entry specifies a description for a cross-certificate type. ([Table 58 on page 583](#) describes the parameters used in cross-certificate type descriptions, except for the Category ID, which is always `xcert`.)

Type ID=`xcert`, Type Name, Type Description

**Table 58:** Parameters in descriptions for cross-certificate types

Parameter	Description
Type ID	<p>Provides a unique representation for the certificate type in the <code>master.certspec</code> file (for example, <code>xcert_default</code>).</p> <p>The characters may be either uppercase or lowercase. Security Manager converts all characters to lowercase when processing the <code>master.certspec</code> file.</p> <p>There is no maximum length for the Type ID, although the first 20 characters must be unique. If the Type ID exceeds 20 characters, it is truncated to 20 characters when Security Manager processes the <code>master.certspec</code> file.</p> <p>It is a good idea to identify the certificate category in the Type ID. For example, you can prefix the Type ID for cross-certificates with <code>xcert_</code>.</p>
Type Name	<p>Defines the name for this certificate type that appears in the list of certificate types in the cross-certification dialog boxes.</p> <p>Note that the name cannot include commas or semicolons. It is limited to 20 characters.</p>
Type Description	<p>Defines the description for this certificate type that appears in the list of certificate types in the cross-certification dialog boxes.</p> <p>Note that the description cannot include commas or semicolons.</p>

For example, here are the values for the parameters that specify one of the default cross-certificate type descriptions:

Type ID	Category ID	Type Name	Type Description
xcert_default	=xcert	,Default	,Default Cross-Certificates

- 3
- Add a comment that describes the purpose of the certificate type when you add a description.

You are not required to add a comment, but you may find them useful. Comments help you remember the purpose of the new information the next time you edit the file. To include a comment, type a semicolon, followed by your comment. When Security Manager processes the file, it ignores everything between the semicolon and the end of the line.

You have now added a certificate type description.

## Editing certificate extensions

Each certificate type for the cross-certificate category contains certificate extensions (see [“What is a certificate extension?” on page 581](#)) in the `master.certspec` file.

The [Extension Definitions] section contains one subsection for certificate extensions that belong to a particular cross-certificate type. You define the subsections in the [Extension Definitions] section for certificate extensions for cross-certificate types just as you define subsections for user certificate extensions. For example, certificate extensions for the `xcert_default` certificate type are defined in the [xcert\_default Verification Definitions] section of the [Extension Definitions] section. For information about creating subsections within the [Extension Definitions] section, see [“Associating a certificate extension with a V1 certificate type” on page 554](#).

---

**Note:** Certificate extensions for cross-certificate types are included only in verification certificates. There is no reason to include these extensions in encryption certificates, which are used to encrypt information.

---

Once you have created a subsection within the [Extension Definitions] section, you can define its certificate extensions.

Topics in this section:

- [“Defining a certificate extension in a subsection” on page 584](#)
- [“Applications of certificate extensions in cross-certificates” on page 585](#)
- [“Limiting trust using distinguished names” on page 586](#)
- [“Limiting trust using policy settings” on page 587](#)
- [“Limiting trust using CA domains” on page 588](#)

### Defining a certificate extension in a subsection

You define a certificate extension within a subsection of the [Extension Definitions] section of the `master.certspec` file by specifying information about the value provided by the extension, and the value itself.

To define a certificate extension, type it in the appropriate subsection of the [Extension Definitions] section. Certificate extensions use the following syntax:

```
Extn Name=Extn OID,Extn Crit,Extn Opt,Extn Type,Extn Value
```

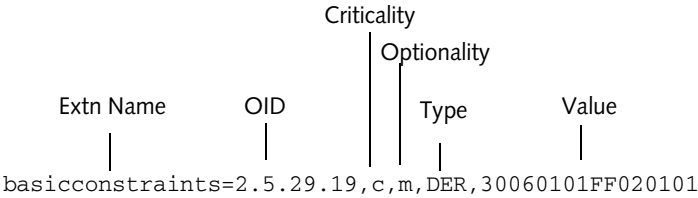
---

**Note:** Security Manager uses the information you provide to encode it as an X.509 certificate attribute. You are not coding an X.509 extension yourself.

---



For example, the following defines a certificate extension named `basicconstraints`. The OID 2.5.29.19 is registered for this certificate extension.



**Note:** You cannot include variable values in certificate extensions for cross-certificate types. You can include only fixed values.

For information about creating certificate extensions, see [“Defining a certificate extension” on page 557](#).

### Applications of certificate extensions in cross-certificates

The value provided by a certificate extension may define a trust limitation between cross-certified CAs. By default, cross-certification enables all users in cross-certified domains to trust each other. For example, Figure 7 shows that Company One’s CA is cross-certified with Company Two’s CA. The users who belong to Company One trust the users who belong to Company Two.

**Figure 7:** Cross-certification of CAs



This default extension of trust is appropriate in many cases. For example, if you have users in different locations (for example, several branch offices), you may use one CA to manage each location. Because you are using different CAs only to manage your user base, you can cross-certify the CAs and you do not need to change the trust extended by default. All users in all CAs can trust each other.

There may be cases, however, when you want to cross-certify CAs, but you do not want to extend trust to all users in the cross-certified CAs. You may want to control which users belonging to a cross-certified CA are trusted. For example, Company One may want to trust only a subgroup of users who belong to Company Two.

Within Security Manager, you can limit trust across cross-certified CAs in three ways. You can limit the trust extended to users in the domain of

- one cross-certified CA based on criteria about user DNs
- one cross-certified CA based on criteria about policy settings in certificates
- a series of two or more cross-certified CAs based on the number of consecutive CAs you want trust extended to

---

**Note:** Trust relationships between cross-certified CAs are defined by two trust models—the hierarchical trust model (see [“Creating subordinate CA certificates” on page 510](#)) and the network trust model (see [“Cross-certifying with other CAs” on page 465](#)). When limiting trust between cross-certified CAs, the methods you use to identify trusted users are the same for both models of cross-certification. However, the relationships between the CAs are different.

---

## Limiting trust using distinguished names

You can limit third-party trust based on user distinguished names (DNs). For users in the domain of a cross-certified CA, you specify that

- only the users whose DNs meet criteria you specify are trusted
- the users whose DNs meet criteria you specify are not trusted

For example, suppose that Company Two identifies the department that a user belongs to in the organizational unit (ou) of the user's DN:

- John Smith works in the Finance department and is assigned the user DN  
“cn=John Smith, ou=Finance, dc=Company Two, dc=com”
- Alice Jones works in the Sales department and is assigned the user DN  
“cn=Alice Jones, ou=Sales, dc=Company Two, dc=com”

If Company Two cross-certifies with Company One, you can limit third-party trust to only the users in the Finance department. Figure 8 shows how you can limit trust between CAs within the network trust model using user DNs. This example shows that users in Company One trust users whose DNs include “ou=Finance” in Company Two's CA. All other users in the domain of Company Two are not trusted.

**Figure 8:** Trust limited by user DN



For the network cross-certification example shown in Figure 8, you can also limit third-party trust to all users in the domain of Company Two, except for users in the Finance department.

You can also use user DNs to limit trust between CAs within the hierarchical trust model. For example, you can specify that users should or should not trust users in the other domain whose DNs include "dc=Company Two, dc=com".

## Limiting trust using policy settings

You can limit third-party trust based on policy settings in certificates. Policy settings used in certificates are created by Entrust PKI administrators and identified using object identifiers (OIDs). OIDs provide a standard mechanism for uniquely identifying things, or "objects," such as certificate policies, encryption algorithms, and directory attributes.

For users in the domain of a cross-certified CA, you can specify that only users and CAs in the path with certificates that contain a policy setting configured to a specific value are trusted. All other users in the domain are not trusted.

For example, suppose that Company Two includes one of the following policy settings in certificates issued to users:

- High assurance
- Medium assurance
- Low assurance

These policy settings indicate how a user proved their identity to the CA before the certificate was issued:

- John Smith identified himself to the CA over the telephone, giving his employee number. His certificate includes the policy setting "Low assurance."
- Alice Jones identified herself to the CA in person, showing her passport. Her certificate includes the High assurance policy setting.

If Company One cross-certifies with Company Two, you can limit third-party trust to the users with certificates in which the Assurance policy setting is configured to either

Medium assurance or High assurance. Figure 9 shows that trust is limited to only users with High assurance.

**Figure 9:** Trust limited by policy settings



You can also use policy settings to limit trust between CAs within the hierarchical trust model. For example, you can specify that users who belong to one CA cannot trust users who belong to another CA if those users hold certificates with the policy setting Low assurance.

---

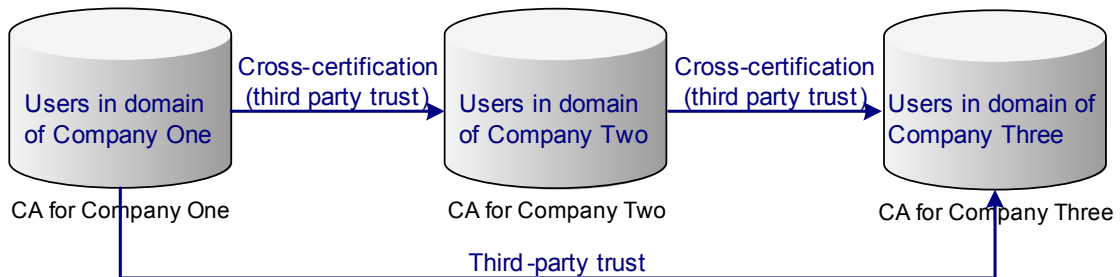
**Note:** If a CA that you are cross-certifying uses a different OID to identify the policy setting, you map the OIDs. In the cross-certificates, you identify the OID that you are using to represent the policy setting and the OID that the other CA is using to represent the same policy setting.

---

## Limiting trust using CA domains

By default, users in the domain of a cross-certified CA also trust all users in all domains across a series of two or more cross-certified CA domains. For example, suppose that your organization is cross-certified with Company Two, and Company Two is cross-certified with Company Three, as shown in Figure 10. By default, all the users in the domain for your organization's CA trust all the users in the domain for Company Two's CA and all the users in the domain for Company Three's CA.

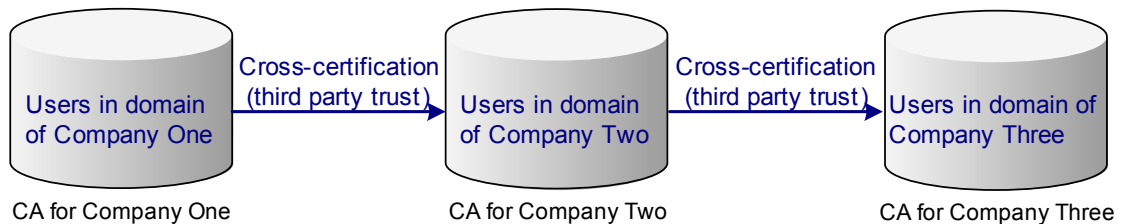
**Figure 10:** Default trust across domains of cross-certified CAs



In some cases, extending third-party trust to all users in the domains of cross-certified CAs may be appropriate. For example, if a different CA manages each department in your organization and all the CAs are cross-certified, you may want third-party trust to exist for all users in all the CA domains.

In other cases, you may want to limit third-party trust to only the users in CA domains for a maximum number of cross-certified CAs. For example, you may want to trust Company Two, but you do not want that trust to extend to Company Three. You can limit third-party trust such that all the users in the domain for your organization's CA trust all the users in the domain for Company Two's CA, but not any of the users in the domain for Company Three's CA (as shown in Figure 11).

**Figure 11:** Limited trust across domains of cross-certified CAs



You can also limit trust between CAs within the hierarchical trust model. For example, you can specify that a CA cannot trust one or more lower-level CAs.

## Examples of cross-certificate types

The following excerpt from an example `master.certspec` file shows the additions to the `master.certspec` file to limit the trust extended to a cross-certified CA.

---

**Note:** The OIDs used in these examples are registered for these certificate extensions.

---



---

**Note:** If you are trying to define `pathlen=0`, the DER representation is `30060101ff020100`.

---

```

[Certificate Categories]
...
[Certificate Types]
...
; -----
; Company One's certificate types
; -----
  
```

```

xcert_sample=xcert,sample,Sample Cross-Certificates
...
[Extension Definitions]
;
[xcert_sample Verification Extensions]

; BasicConstraints with cA = TRUE, pathLenConstraint = 1
basicconstraints=2.5.29.19,c,m,DER,30060101FF020101
; Certificate policies with policyIdentifier = 1.1.1.1
certificatepolicies=2.5.29.32,c,m,DER,300730050603290101

; Policy mappings mapping 1.1.1.1 to 1.1.1.2
policymappings=2.5.29.33,n,m,DER,300C300A06032901010603290102

; Policy constraints with requireExplicitPolicy = 0
policyconstraints=2.5.29.36,c,m,DER,3003800100
...

```

# Customizing CA certificates

Certificates issued from the CA certificate category are used for two purposes:

- CA verification certificates contain the CA verification public key used to verify the CA signature on users' certificates.
- CA link certificates provide a connection between CA key pairs once they are updated.

Security Manager issues a CA verification certificate when you initialize Security Manager after installation. Security Manager issues CA link certificates when CA key pairs are updated. For information about CA verification and CA link certificates, see the *Security Manager Operations Guide*.

You can define additional content that you want Security Manager to include in CA certificates. Security Manager includes this additional content when issuing certificates for the certificate type.

---

**Note:** You cannot create new CA certificate types. You can only define the additional content that you want Security Manager to include in certificates issued for the existing types.

---

The certificate types in the CA certificate category that Security Manager issues are defined in the `master.certspec` file. The following excerpt from the default `master.certspec` file shows the descriptions for the predefined CA certificate types.

```
[Certificate Categories]
...
[Certificate Types]
...
ent_default=enterprise,Default,Default Enterprise Certificates
web_default=web,Default,Default Web Certificates (for browsers and secure email)
xcert_default=xcert,Default,Default Cross-Certificates
polcert_cliset=policycert,Client Settings,Client Setting Certificates
cacert_default=cacert,Default,Default Self Signed CA Certificates
cacert_link=cacert,Link,Self Issued Link CA Certificates
...
```

For example, here are the values for the parameters that specify the description for the predefined CA verification certificate type:

Type ID	Category ID	Type Name	Type Description
cacert_default	cacert	Default	Default Self Signed CA Certificates

Extra content for CA certificates is defined using certificate extensions. The `cacert_default` certificate type does not have any predefined certificate extensions.

The `cacert_link` certificate type has two predefined certificate extensions. These extensions are required for certificates to comply with RFC 2549. The following excerpt from the `master.certspec` file shows the certificate extensions.

```
[Certificate Categories]
...
[Certificate Types]
...
cacert_link=cacert,Link,Self Issued Link CA Certificates
...
[Extension Definitions]
;
[cacert_link Verification Extensions]
basicConstraints=2.5.29.19,c,m,DER,30030101FF;BasicConstraints w/cA = TRUE
certificatePolicies=2.5.29.32,n,m,DER,300830060604551d2000;
...
```

The following shows values for the parameters that specify one of the default certificate extensions for the CA link certificate type description:

Extn Name	OID	Criticality	Optionality		Value
			Type		
basicConstraints	2.5.29.19	c,m	DER		30030101FF

For information about defining certificate extensions for CA certificates, see [“Editing certificate extensions” on page 553](#).



# Excluding certificate extensions by certificate type

Certificates issued by Security Manager are compliant with the X.509 recommendation (see [“About the X.509 recommendation” on page 528](#)). This recommendation specifies the required certificate extensions for X.509-compliant certificates. Security Manager inserts the values provided by these X.509 extensions in certificates when they are issued.

The size of a certificate depends on the names and extensions defined in the certificate. You can specify that you want Security Manager to exclude certain X.509 certificate extensions. This is especially valuable when you want to reduce your certificate storage requirements, for example, if you plan to deploy WAP (Wireless Application Protocol) servers and devices.

You can exclude certain certificate extensions from particular certificate types, as described in this section. However, you can also exclude certain extensions on a global basis with advanced settings. You can use advanced `entmgr.ini` file settings prior to initializing Security Manager (see the *Security Manager Operations Guide*), or you can use advanced Security Manager Control Command Shell settings after initialization (see the *Security Manager Operations Guide*).

---

**Note:** If you exclude any of the extensions from the X.509 recommendation, your certificates may no longer be compliant with this recommendation.

---

Topics in this section:

- [“Extensions you can exclude from user certificates” on page 594](#)
- [“Extensions you can exclude from CA certificates” on page 594](#)
- [“Extensions you can exclude from cross-certificates” on page 595](#)
- [“Excluding default extensions from an X.509 certificate” on page 595](#)

## Extensions you can exclude from user certificates

There are eight default extensions that Security Manager includes in every certificate issued from the Enterprise and Web categories. Table 59 lists these extensions and indicates which of the extensions you can exclude for certificate types in the Enterprise and Web categories.

**Table 59:** Removable default user certificate extensions

Extension	Removable
Basic Constraints	Yes <sup>1</sup>
Private Key Usage Period	Yes <sup>2</sup>
Authority Key Identifier	No
Subject Key Identifier	Yes <sup>3</sup>
CRL Distribution Points	Yes <sup>4</sup>
Entrust Version Information	Yes (not recommended) <sup>5</sup>
Key Usage	No
Subject Directory Attributes <sup>6</sup>	No

1. You can also exclude this extension from all user certificates with a Command Shell setting.
2. You can also exclude this extension from all user certificates with a Command Shell setting.
3. This extension can be excluded only if you are working with Entrust desktop applications built on release 5.1 or later Entrust toolkits or with Entrust Entelligence Desktop Solutions 5.1 or later. If you are working with older clients, do not exclude this extension.
4. If you exclude this extension, you should enable the combined CRL. See the *Security Manager Operations Guide*.
5. It is recommended that you do not exclude the Entrust Version Information extension for Enterprise certificates. This extension contains information needed to update a user's keys. In particular, do not exclude this extension for Entrust client 5.x users who have key expiry dates set. This practice can result in the client attempting a key update at each login.
6. This extension is inserted in certificates only for Enterprise users who are not in the End User role.

## Extensions you can exclude from CA certificates

Security Manager inserts eight extensions by default in CA certificates. Table 60 lists these extensions and indicates which of the default extensions you can exclude for certificate types in the CA category.

**Table 60:** Removable default CA certificate extensions

Extension	Removable
Basic Constraints	Yes (not recommended) <sup>1</sup>
Private Key Usage Period	Yes <sup>2</sup>

**Table 60:** Removable default CA certificate extensions (continued)

Extension	Removable
Authority Key Identifier	Yes for root CA certificates No for Link certificates
Subject Key Identifier	No
CRL Distribution Points	Yes <sup>3</sup>
Entrust Version Information	Yes
Netscape Certificate Type	Yes
Key Usage	Yes

1. The basicConstraints extension is required for root CA and cross-certificates. Excluding this extension from this certificate type can cause problems with Security Manager and client applications. It is recommended that you do not exclude this extension.
2. You can also exclude this extension from all certificates in the CA category with a Command Shell setting.
3. Excluding the cRLDistributionPoints extension from a certificate indicates that the certificate will be revoked on the combined CRL. Exclude the cRLDistributionPoints extension only if the combined CRL is enabled. The combined CRL is enabled if either of the advanced settings UseCombinedCRL or MsCompatibility are 1. For information about these advanced settings, see the *Security Manager Operations Guide*.

You may exclude the cRLDistributionPoints extension from the CA root certificate if the combined CRL is not enabled as long as you are sure that relying parties will not be checking the revocation of the root CA. If you exclude the cRLDistributionPoint extension from the CA root certificate and the combined CRL is not enabled, extra steps are required to revoke the CA root certificate. See the *Security Manager Operations Guide* for information about revoking CA root certificates.

## Extensions you can exclude from cross-certificates

Security Manager inserts six extensions by default in cross-certificates. Table 61 lists these extensions and indicates which of the default extensions you can exclude for certificate types in the cross-certificate category.

**Table 61:** Removable default cross-certificate extensions

Extension	Removable
Basic Constraints	Yes (not recommended) <sup>1</sup>
Authority Key Identifier	No
Subject Key Identifier	No <sup>2</sup>
CRL Distribution Points	Yes <sup>3</sup>
Entrust Version Information	Yes
Key Usage	Yes

1. The basicConstraints extension is required for cross-certificates. Excluding this extension from this certificate type can cause problems with Security Manager and client applications. It is recommended that you do not exclude this extension.
2. When you are cross-certifying two CAs, they must use identical key identifier modes. See [“Key identifiers in cross-certification” on page 470](#).
3. If you exclude this extension, you should enable the combined CRL. See the *Security Manager Operations Guide*.

## Excluding default extensions from an X.509 certificate

You edit the certificate specifications file to exclude default extensions from an X.509 certificate.

- If you want to exclude default extensions from the Default Enterprise or Default CA certificate type, see [“To exclude default extensions from the Default Enterprise or Default CA certificate type” on page 596](#).
- If you want to exclude extensions from a certificate type other than the Default Enterprise or Default CA certificate type, see [“To exclude default extensions from certificate types other than Default Enterprise or Default CA” on page 597](#).

### To exclude default extensions from the Default Enterprise or Default CA certificate type

- 1 In Security Manager Administration, click **File > Certificate Specifications > Export**.

By default, the certificate specifications file is called `master.certspec`; however, this file may be renamed during installation of Security Manager.

Save the file to a known location on disk.

- 2 Open the file in a text editor.
- 3 Find the [Advanced Settings] section.
- 4 To exclude an extension from the existing Default Enterprise certificate type, find the [ent\_default Advanced] section and remove the semicolon (;) that precedes the advanced setting that corresponds to the extension you want to exclude.

To exclude an extension from the existing Default CA certificate type, find the [cacert\_default Advanced] section and remove the semicolon (;) that precedes the advanced setting that corresponds to the extension you want to exclude. Uncomment the following entries to exclude the corresponding extensions:

Extension	To exclude, uncomment this line
Basic Constraints	noBasicConstraints=1

Extension	To exclude, uncomment this line
Private Key Usage Period	noPrivateKeyUsage=1
Authority Key Identifier	noAuthorityKeyId=1
CRL Distribution Points	noCRLDistPoints=1
Entrust Version Information	noEntrustVersInfo=1
Key Usage	noKeyUsage=1
Subject Key Identifier <sup>1</sup>	noSubjectKeyId=1

1. By default, "noSubjectKeyId=1" does not appear in the initial.certspec or master.certspec file. This setting is intended for use with Entrust desktop applications built on release 5.1 or later Entrust toolkits or for use with Entrust Entelligence Desktop Solutions 5.1 or later. If you are working with older clients, do not add this advanced setting to the initial.certspec or master.certspec file.

---

**Note:** Uncommenting the line `noPrivateKeyUsage=1` also means that the CA private key usage period is set to 100% of the CA verification certificate lifetime. Enabling `noPrivateKeyUsage=1` in the `master.certspec` file will override any value set during Security Manager configuration or subsequently set in Security Manager Control Command Shell.

---

- 5 Click **File > Save** and close the file.
- 6 In Security Manager Administration, click **File > Certificate Specifications > Import** to import the updated file.

Make sure that you select the same file that you just edited.

You have now excluded default extensions from an X.509 certificate.

### To exclude default extensions from certificate types other than Default Enterprise or Default CA

- 1 In Security Manager Administration, click **File > Certificate Specifications > Export**.

By default, the certificate specifications file is called `master.certspec`; however, this file may be renamed during installation of Security Manager.

Save the file to a known location on disk.

- 2 Open the file in a text editor.
- 3 Find the [Advanced Settings] section.
- 4 In the [Advanced settings] section, type a section header using the following format:

[<Type ID> Advanced]

where <Type ID> is the identifier of the certificate type as defined in the [Certificate Types] section. For example, the following header identifies certificate extensions to exclude from the Link CA certificate type:

```
[cacert_link Advanced]
```

- 5** On a new line in this section, type the entry corresponding to the extension you want to exclude, and set its value to 1.

**Attention:** Before you exclude any extension, check [Table 59 on page 594](#) through [Table 61 on page 595](#) to ensure that you can exclude the extension for the certificate type.

Add the following entries to exclude the corresponding extensions:

Extension	To exclude, add this line
Basic Constraints	noBasicConstraints=1
Private Key Usage Period	noPrivateKeyUsage=1
Authority Key Identifier	noAuthorityKeyId=1
CRL Distribution Points	noCRLDistPoints=1
Entrust Version Information	noEntrustVersInfo=1
Key Usage	noKeyUsage=1
Subject Key Identifier <sup>1</sup>	noSubjectKeyId=1

1. By default, “noSubjectKeyId=1” does not appear in the initial.certspec or master.certspec file. This setting is intended for use with Entrust desktop applications built on release 5.1 or later Entrust toolkits or for use with Entrust Entelligence Desktop Solutions 5.1 or later. If you are working with older clients, do not add this advanced setting to the initial.certspec or master.certspec file.

The following example shows the exclusion of the Key Usage extension from certificates issued for the Link CA certificate type:

```
[cacert_link Advanced]
noKeyUsage=1
```

- 6** Click **File > Save** and close the file.
- 7** In Security Manager Administration, click **File > Certificate Specifications > Import** to import the updated file.

Make sure that you select the same file that you just edited.

You have now excluded default extensions from an X.509 certificate.

# Making a certificate type obsolete

If your organization no longer uses a certificate type, you cannot delete it even if no users are enabled with that type. Instead, you make the certificate type obsolete.

You make a certificate type obsolete by defining its replacement certificate type. If Security Manager receives a request to issue a certificate for a type that is defined as obsolete (for example, when a certificate requires updating), Security Manager uses the replacement certificate type.

You can make a certificate type obsolete if you (or another Security Officer) created it. You can also make the Timestamp Server certificate type, the Roaming Server certificate type, and the sample Web certificate types obsolete.

---

**Attention:** You cannot undo the action once you make a certificate type obsolete.

You cannot rename or delete certificate definitions. If you make the certificate type obsolete by replacing it with a new certificate type, you must copy the certificate definitions from the obsolete certificate type to the replacement.

If your certificate type creates V2 key pairs, you cannot make it obsolete. You can only make certificates with V1 key pairs obsolete.

---

## Defining a replacement certificate type

You make a certificate type obsolete by defining its replacement certificate type in the `master.certspec` file. For information about creating, opening, and processing the `master.certspec` file, see [“Working with Security Manager certificate specifications” on page 531](#).

If the replacement certificate type includes a certificate extension with a variable value that is mandatory, Security Manager cannot issue the certificate (unless a value for the variable is already provided). The Entrust PKI administrator must re-enable each user and provide values for the mandatory certificate extension in the replacement certificate type.

When you are defining a replacement certificate type, it must contain certificate definitions with the same names as those in the obsolete certificate type it is replacing. However, the replacement certificate type can also include additional certificate definitions.

### To make a certificate type obsolete

- 1 In the `master.certspec` file, locate the [Obsolete Certificate Types] section.
- 2 Define a certificate type as obsolete using the following syntax:

<Obsolete Type ID>=<Replacement Type ID>

where

- <Obsolete Type ID> identifies the certificate type that you want to make obsolete (as defined in the [Certificate Types] section).
- <Replacement Type ID> identifies the certificate type (as defined in the [Certificate Types] section) that you want to replace with the obsolete one.

For example, the following excerpt from an example `master.certspec` file shows that the sample certificate type, named `web_server`, is obsolete. The default Web certificate type, named `web_default`, replaces the `web_server` certificate type.

```
[Certificate Categories]
...
[Obsolete Certificate Types]
web_server=web_default
...
```

---

**Note:** It is a good idea to include a comment that describes why you made a certificate type obsolete. Comments help you remember the reason the next time you edit the file. To include a comment, type a semicolon followed by your comment. When Security Manager processes the file, it ignores everything between the semicolon and the end of the line.

---

Once you define a certificate type as obsolete, you can remove its certificate extensions or certificate attributes and variables from the `master.certspec` file. To remove certificate extensions or certificate attributes and variables for an obsolete certificate type, select the appropriate lines and delete them.



# Customizing database fields

You can add custom information about users to the Security Manager database. You add this information using database fields.

Database fields are associated with user certificate types (certificate types in the Enterprise and Web certificate categories). Security Manager Administration prompts for the information for a database field when an Entrust PKI administrator enables a user using that certificate type. The database fields associated with a certificate type are listed in the **New User** dialog box below the list of certificate types.

The 'New User' dialog box is shown with the 'Certificate Info' tab selected. The 'Category' dropdown is set to 'Enterprise'. The 'Type' list displays the following certificate types and descriptions:

Certificate Type	Description
Default	Default Enterprise Certificates
Timestamping Agent	Timestamping Agent Certificates
Timestamping Agent Citi...	Timestamping Agent Certificates
Roaming Server	Roaming Server Certificates
Export	Exportable Enterprise Certificates
Desktop Admin	Desktop Admin Certificates
TruePass Server	TruePass Server Certificates
TruePass Server Multido...	TruePass Server Multidomain Primar
IPSec Device	IPSec device certificates stored in d

Asterisks (\*) appear beside required extensions.

Under the 'Certificate Information' section, there are two input fields:

- Department number: [Text Field]
- \* Pre-shared secret: [Text Field]

At the bottom of the dialog are 'OK', 'Cancel', and 'Help' buttons.

Database fields are useful for storing information about users when you want to limit the people who have access to that information. For example, you can use database fields to store a secret provided by the user. The user must tell this secret to an Entrust PKI administrator before obtaining sensitive information through an unsecured method of communication (for example, the telephone). The information provided for the database fields is stored only in the Security Manager database. Although database fields are associated with particular certificate types, the information is not inserted in a certificate.

You can create database fields if you are a Security Officer or if you have the appropriate permissions (see [“Administering roles” on page 353](#)). You create database fields using the `master.certspec` file.

**Note:** This section assumes that you know how to create, open, and process the `master.certspec` file. If you do not know how to perform these tasks, see [“Working with Security Manager certificate specifications” on page 531](#).

Topics in this section:

- [“Defining a database field” on page 602](#)
- [“Example of database fields” on page 603](#)

## Defining a database field

You define database fields for a particular user certificate type. If the certificate type does not exist in the `master.certspec` file, you must define it. For information about defining a certificate type, see [“Editing a certificate type description” on page 549](#).

You define database fields in the [Database Field Definitions] section of the `master.certspec` file. A subsection header identifies the certificate type that database fields are associated with. For example, the following section header in the [Database Fields] section identifies database fields associated with the `ent_default` certificate type:

```
[ent_default Database Fields]
```

Each database field definition uses the following format. (Table 62 describes the parameters used in database field definitions.)

```
field_id=secrecy,optionality,<variable_id>
```

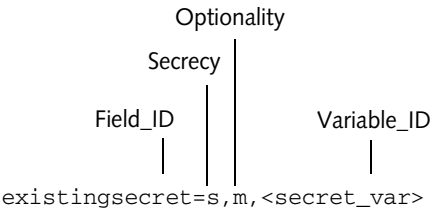
**Table 62:** Parameters in database field definitions

Parameter	Description
field_id	Identifies the field within the <code>master.certspec</code> file.
secrecy	Indicates whether the field is secret. The value is <code>s</code> (secret) or <code>n</code> (not secret).  Values for secret database fields are encrypted in the Security Manager database.  Values for non-secret database fields are published in a database table named <code>CustomUserInfo</code> . Each non-secret value is stored in its own row in this table, along with the user ID that corresponds to the user who is assigned this value, and the <code>field_id</code> . Using a third-party reporting tool, you can generate reports that include values for non-secret database fields.

**Table 62:** Parameters in database field definitions (continued)

Parameter	Description
optionality	Indicates whether you must provide a value for the field in the New User dialog box. The value is <code>o</code> (optional) or <code>m</code> (mandatory).  An Entrust PKI administrator is required to enter values only for fields that are mandatory. These fields are identified by an asterisk in the <b>New User</b> dialog box.
variable_id	Names the variable that represents the value. You include only one variable for each database field.  You must define the variable in the <code>[Variables]</code> section of the <code>master.certspec</code> file. You define variables for database fields just as you define variables for certificate extensions and certificate attributes. For information about defining variables, see <a href="#">“Editing a variable” on page 565</a> .

For example, the following defines a database field named `existingsecret`:



## Example of database fields

The following example shows the additions to the `master.certspec` file to create three database fields for Enterprise certificates issued for the `ent_default` certificate type:

- a preshared secret that must remain confidential
- charge-back information that you can include in a report
- a comment pertaining to the user that you can include in a report

The comment information is optional, but the preshared secret and the chargeback information are required.

The following excerpt from the `master.certspec` file shows the information you enter to define these database fields:

```
[Certificate Categories]
...
[Certificate Types]
...
```

ent\_default=enterprise,Default,Default Enterprise Certificates

...

[Database Field Definitions]

[ent\_default Database Fields]

existingsecret=s,m,<secret\_var>

chargeback=n,m,<chargeback\_var>

comment=n,o,<comment\_var>

.

[Variables]

secret\_var=TextString,Pre-shared Secret,Secret information which identifies the user.,Range,1,20

chargeback\_var=TextString,Charge-back Dept.,Department to be charged for this user's certificates.,Range,6,6

comment\_var=TextString,Comment,Miscellaneous information about the user.,Range,1,50

...

# Turning off revocation checking for foreign certificates

Entrust PKI clients (such as Security Manager Administration and Entrust Entelligence Desktop Solutions) rely on the directory for revocation information in the form of revocation lists. When the subordinate CA certificate is issued by the superior CA, Entrust clients retrieve the revocation list issued by the superior CA to determine whether a subordinate CA certificate is revoked.

In certain environments, the clients may be unable to obtain the revocation list. This may happen for either of the following reasons:

- The superior CA does not publish revocation lists to the directory.
- The superior CA's directory is not accessible to the clients.

Since the clients do not have access to revocation lists issued by the superior CA, the clients is unable to validate the subordinate CA certificate and hence will not be able to validate certificate chains that contain the subordinate CA certificate.

Revocation checking for foreign certificates can be turned off. From an end-entity's viewpoint, foreign certificates are those that are not issued by the CA that issued its certificates.

## To turn off revocation checking

- 1 To turn off revocation checking, you need to add an attribute to the policy certificate.

To customize the policy certificate, you edit the `master.certspec` file (or the `initial.certspec` file). Before you can edit this file, you need to export it. In Security Manager Administration, click **File > Certificate Specifications > Export**.

- 2 Place this setting in the `[polcert_cliset Attributes]` section:

```
;
;-----
; Disable RL Checking for Foreign Certificates
;-----
check_rls=1.2.840.113533.7.77.18,Boolean,0
```

---

**Attention:** You cannot enable the XAP subsystem if foreign revocation list checking is disabled.

---

- 3 Save your changes and import the revised `master.certspec` file. In Security Manager Administration, click **File > Certificate Specifications > Import**.
- 4 Once you successfully import the `master.certspec` file, test your change.

After you successfully import the `master.certspec` file, Security Manager should issue the following audit if the services are running:

```
[-07732 User Policy updated.]
```

The detailed audit message lists all the settings that went into the policy certificate and one of them should be

```
check_qls id-Entrust-SecureNetworks.77.18 010100
```

You have now turned off revocation checking.

## Working with audit logs and creating reports

Using Security Manager Administration, Entrust PKI administrators can monitor Security Manager activity and create general or highly detailed reports about this activity.

This chapter includes the following sections:

- [“Viewing Security Manager Administration log files” on page 608](#)
- [“Working with audit logs” on page 609](#)
- [“Producing reports in Security Manager Administration” on page 616](#)
- [“Creating reports” on page 619](#)
- [“Report fields” on page 629](#)

# Viewing Security Manager Administration log files

Security Manager Administration creates informational log files called `DirOperationsDDMMYY-HHMMSS.log` (for example, `DirOperations280504-142535.log`) that contain any directory operations performed on Security Manager Administration on that particular machine. By default, these files are located in `C:\Program Files (x86)\Entrust\Security Manager Administration\data` by default.

See [“Working with audit logs” on page 609](#) for additional information on Security Manager Administration log files.



# Working with audit logs

Audit logs record all transactions that involve the Certification Authority (CA), such as Entrust PKI administrator logins and logouts, key updates, user-related activities such as user creation, changes to roles, groups, or searchbases, and any system failures. The audit logs include audit logs generated by XAP clients such as Administration Services.

The audit logs also record all EAC transactions related to the CVCA or DV, such as certificate issuance and key updates. For more information about EAC, see the *Entrust ePassport Solutions Guide*.

Use audit logs to troubleshoot the system and generally monitor user activity. You can only view audit logs if your role includes sufficient permissions. For more information about roles and permissions, see [“Administering roles” on page 353](#).

Topics in this section:

- [“Viewing audit logs” on page 609](#)
- [“Viewing audit log details” on page 611](#)
- [“Description of audit log fields” on page 612](#)
- [“Sorting audit logs” on page 613](#)
- [“Clearing audit logs” on page 614](#)
- [“Saving audit logs” on page 614](#)

## Viewing audit logs

You can view audit logs in Security Manager Administration as described in the following procedure. Alternatively, you can view audit logs using the Security Manager Control Command Shell (see the *Security Manager Operations Guide*).

You can only view audit logs if your role includes sufficient permissions. For more information about roles and permissions, see [“Administering roles” on page 353](#).

### To view audit logs

- 1 Log in to Security Manager Administration as an Entrust PKI administrator. See [“Logging in to Security Manager Administration” on page 46](#).  
Security Manager Administration appears.
- 2 Click **Audit Logs > Search**.

The **Audit Search** dialog box appears.

**Audit Search**

Time Range

From: Date: 10 - 11 - 2010 Time: 11 : 38 : 29 AM

To: 10 - 12 - 2010 11 : 38 : 29 AM

☒ Search to current date and time

Additional Search Criteria

String:

Audit Number:

Severity: < Any >

OK Cancel Help

- 3** Set the **Date** and **Time** in the **From** and **To** fields. Select **Search to current date and time** to include the most recent logs in your search.  
You can choose one-day and one-minute increments, or highlight the date and time and type new values.
- 4** Type a text string in the **String** field to find audit logs that include such text in any of the audit log fields (see [Table 63 on page 612](#)).  
If you are not looking for a particular text string, use the wildcard (\*) setting.
- 5** Type a log number in the **Audit Number** field, to find a particular event log number.  
If you are not looking for a particular log number, use the wildcard (\*) setting.
- 6** To search for logs based on a particular severity, choose **ALARM**, **Event**, or **Log** in the **Severity** drop-down list.  
If you are not looking for a particular type of log, use the default **< Any >** setting.

---

**Note:** You may customize the severity rating of audit records. If your organization has customized the severity rating of any of the audit records, the severity rating that appears in Security Manager Administration may not match the severity rating that appears in your Entrust log files. For details, see the *Security Manager Operations Guide*.

---

**7** Click **OK**.

A dialog box appears indicating successful completion of the operation. The Audit Log Entries property page displays the audit logs.

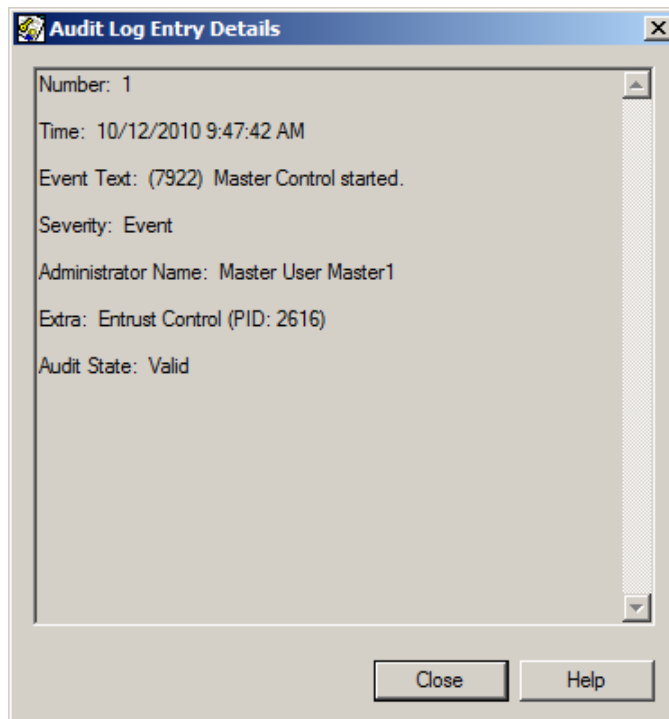
## Viewing audit log details

You can view all the information about an audit log in the Audit Log Entry Details dialog box.

### To view audit log details

- 1 Search for a range of audit logs as described in [“To view audit logs” on page 609](#).
- 2 Double-click an audit log.

The **Audit Log Entry Details** dialog box appears.



You have now viewed an audit log in detail.

## Printing audit logs

There is no print option available with the audit logs; however, you can save your audit report to a text file and then use a text editor to print the report.

### To print audit logs

- 1 Search your audit logs as described in [“Viewing audit logs” on page 609](#).
- 2 Save the search results to a text file as described in [“Saving audit logs” on page 614](#).
- 3 Open the text file and use the text editor’s print command.

## Description of audit log fields

Table 63 lists the log fields that appear in the **Audit Log Entries** property page. The *Security Manager Operations Guide* lists all audits and a brief description of each, and describes what actions you must take when certain audits occur.

**Table 63:** Description of audit log fields in Security Manager Administration

Field	Description
Num	Displays the unique audit log number, which increments by one for each log, starting at 1.
Time	Displays the date and time when an audit event occurred. The time is based on the time zone of the machine the administrator is logged into.
Event Text	Displays a brief explanation of what event the audit log is related to, along with an audit identifying number. See the <i>Security Manager Operations Guide</i> for audit descriptions.
Severity	Displays one of three severity levels, as follows: <ul style="list-style-type: none"><li>• Log is the least severe level.</li><li>• Event is the next severity level.</li><li>• ALARM requires the attention of a Master User.</li></ul>
Administrator Name	Displays the DN of the Entrust PKI administrator that performed this action.
Target Name	Refers to the entity to which the action reported by the audit was applied. If the action does not apply to a particular entity, this field is does not display.
Extra	May show additional information about the audit.

**Table 63:** Description of audit log fields in Security Manager Administration (continued)

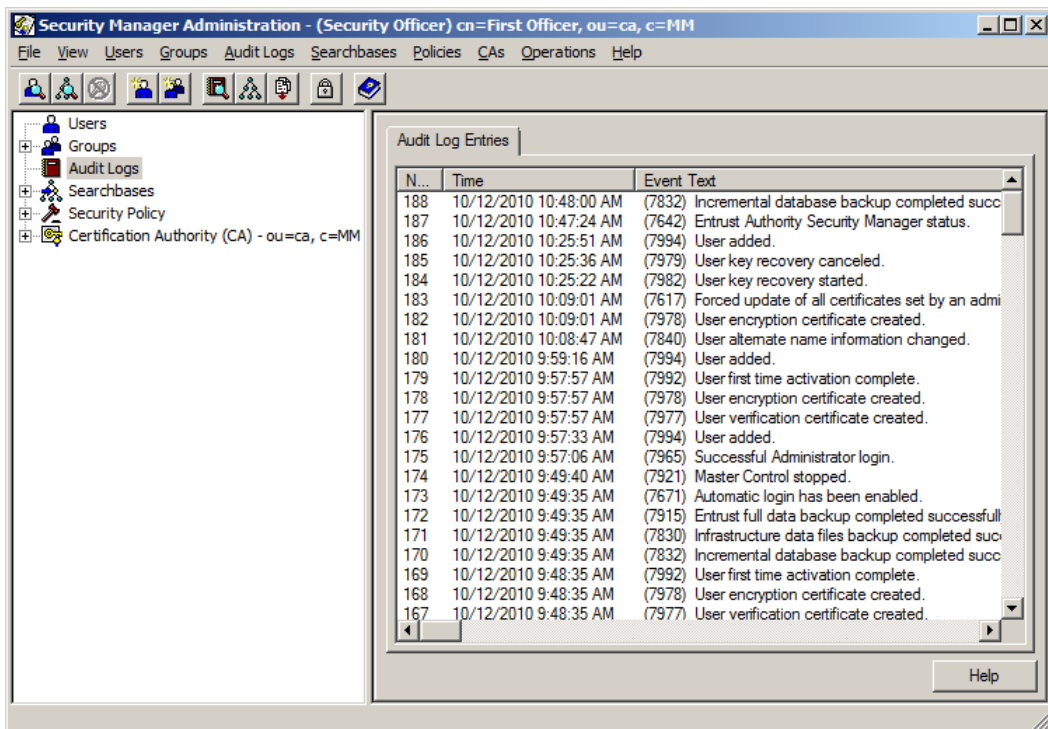
Field	Description
Audit State	<p>Shows whether an audit log is valid, according to a message authentication code (MAC), which guarantees that it has not been modified since the log was created.</p> <p>Audit states are as follows:</p> <ul style="list-style-type: none"><li>• <b>Valid.</b> The audit log is valid. Note that all audit states other than <b>Valid</b> suggest tampering with the audit log file, or corrupt data. Any audit state other than <b>Valid</b> indicates a situation requiring investigation by a Master User.</li><li>• <b>Invalid.</b> The audit log is corrupt.</li><li>• <b>Missing.</b> The audit log is missing from an audit log file.</li><li>• <b>Out of Range.</b> The audit log unique number is outside the range specified in the audit log file header.</li><li>• <b>Parse Error.</b> The encoded audit log could not be decoded. This audit log is corrupt.</li><li>• <b>Out of Sequence.</b> One or more audit logs were found to be out of sequence. For example, audit log 55 was found after audit log 60.</li><li>• <b>Bad File Header.</b> The audit log file header could not be read; in this case none of the audit logs in the audit log file can be validated.</li></ul>

## Sorting audit logs

After you search for audit logs, you can sort the audit logs by one of several list categories. For example, to sort the audit logs into groups of Logs, Events, and ALARMS, click **Severity**. The audit logs display beginning with the most severe category, ALARM, and end with the least severe category, Log.

### To sort audit logs

- 1 Search for audit logs as described in [“To view audit logs” on page 609](#).  
The **Audit Log Entries** property page displays the audit logs.



2 Click a title bar button to sort the logs accordingly.

You have now sorted the audit logs.

## Clearing audit logs

After you search, sort, and view audit logs in the **Audit Log Entries** property page, you can clear the property page to perform new searches. To do this, click **Audit Logs > Clear Audit Logs List**.

## Saving audit logs

As an Entrust PKI administrator with sufficient permissions, you can save audit logs to a location on the machine hosting Security Manager Administration. Audit logs are saved as text files.

You can open the text files later in a spreadsheet application where you can sort audit logs as you can in Security Manager Administration. Note that text files are not secured as they are not encrypted and signed. Anyone can open them in a spreadsheet application.

### To save audit logs

- 1 Search for audit logs as described in [Step 1](#) through [Step 3](#) of “[To view audit logs](#)” on page 609.
- 2 Click **Audit Logs > Save As**.
- 3 Select the location, a name, and an extension; or you can accept the default and click **OK**.

You have now saved an audit log.

## Creating a debug log

If you are trying to troubleshoot a problem with Security Manager Administration, you can configure the audit process to record debug information.

### To create a debug log

- 1 Open the `entrust.ini` file for Security Manager Administration in a text editor
- 2 In the `[ASH Information]` section, add an entry similar to this:  
`DebugFile=c:\temp\adminlog.txt`
- 3 Save and close the file.
- 4 Restart Security Manager Administration for the change to take effect.
- 5 Repeat the operation that caused your problem.
- 6 Check the `adminlog.txt` file. It now contains additional information to help you troubleshoot your problem.

# Producing reports in Security Manager Administration

Security Manager Administration has a reporting feature that lets you create a list of all users in your Certification Authority (CA) domain, or lists of users in various states or with various characteristics, within a certain time period.

---

**Note:** If you are producing a large report, you may need to increase the permitted size of the ASH service message. To do so, increase the `MaxMessage` setting value in the `[ASH Information]` section of the `entrust.ini` file. For more information, see the *Security Manager Operations Guide*.

---

Topics in this section:

- [“Report contents” on page 616](#)
- [“Report formats” on page 617](#)

## Report contents

Security Manager Administration allows you to tailor the scope of your report by defining users' characteristics. These characteristics are divided into three different areas:

- General information  
This group of characteristics includes the current state, role, and group of the users, and whether they have attribute certificates or activation codes.
- Certificate information  
This group of characteristics includes the certificate category and type of the users, and whether they have revoked, expired, or issued certificates. Certificate information characteristics also allow you to include users with lifetime information in their policy certificates, or users with a key update pending.
- State history  
This group of characteristics defines the state of the users as added, active, deactivated, in key recovery, or export.

Each group corresponds to a property page of the **Create Report** dialog box, which you can use to define the scope of the report. These property pages are identical to those used to define the search criteria when you are searching for users by Entrust properties.

As well as defining the users' characteristics, you can determine the time period to affect the report through a separate property page. This means that you can combine



the general, certificate, and state parameters with a range of time to narrow your report criteria. For example, you can specify a report on

- active users who have attribute certificates
- users who were exported last year
- users with certificates that were issued during a particular month last year
- users whose certificates expired in the last week

When you create a report, Security Manager Administration ANDs together all the parameters you selected, and searches for the users who meet all the criteria during the time period you specified.

See [“Finding users by Entrust properties” on page 134](#) for detailed information about the fields on each property page.

## Report formats

Security Manager Administration allows you to generate highly detailed and flexible reports in both XML and tab-delimited formats, using the **Report Type** property page.

Topics in this section:

- [“XML-format reports” on page 617](#)
- [“Tab-delimited reports” on page 618](#)

For more information on what fields can appear in a report, see the section [“Report fields” on page 629](#).

### XML-format reports

If you choose the XML format, Security Manager Administration produces an XML file that contains the information you selected. You can specify that this file contains

- all user property information
- all user status information
- activation codes and dates, or just the dates
- registration password information
- user certificate variables and database fields
- certificate information for all certificates or just the latest certificates, information for pending certificates, or the certificate's PEM-encoded data

If any of this information is not necessary, you can leave the corresponding property page fields unchecked to avoid releasing sensitive information or to restrict the file size.

The XML format also allows you to specify a stylesheet if your browser supports them. (Consult your browser documentation for more information on XML stylesheets.) Security Manager provides a sample stylesheet called

samplereport.xml in the <Install\_Dir>\etc directory. The sample stylesheet converts an XML-format report into an HTML table that contains a subset of the XML information. XML reports are saved in XML files in a folder of your choice.

---

**Note:** When generating XML reports larger than 1 MB, you need to reset the maximum size of the message. To do so, edit the `MaxMessage` setting in the [ASH Information] section of the `entrust.ini` file. For more information on this setting, see the *Security Manager Operations Guide*.

---

## Tab-delimited reports

If you choose the tab-delimited format, Security Manager Administration produces a tab-delimited text file that you can view using any spreadsheet or text editor application. These reports are saved in text files in a folder of your choice.

# Creating reports

The following procedure explains how you can create a report using Security Manager Administration. You must have the **Create Reports** permission to create reports.

---

**Note:** If you create a report of all Entrust users, every former user is listed, even if you have disabled and deleted the user entries in the directory. This occurs because Security Manager keeps a key history of all users in the database so that it can still decrypt files even after the users leave the organization.

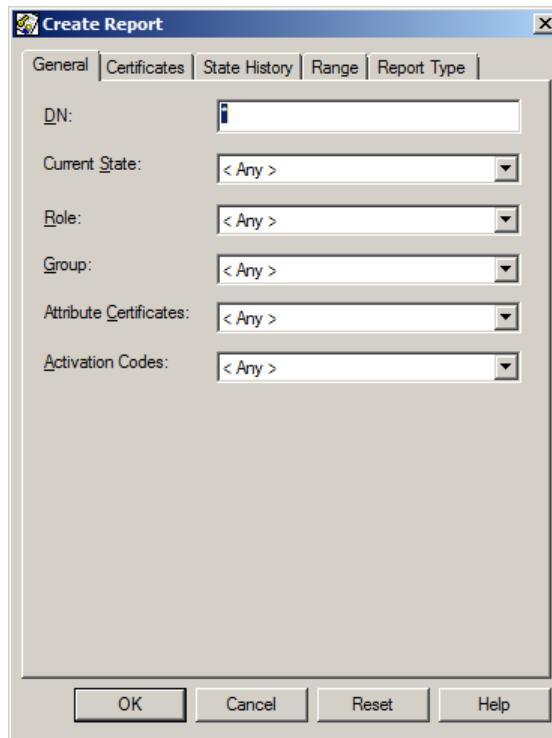
---

## To create a report

- 1 Log in to Security Manager Administration as an Entrust PKI administrator (see [“Logging in to Security Manager Administration” on page 46](#)). Security Manager Administration appears.

- 2 Click **Operations > Create Report**.

The **Create Report** dialog box appears.



---

**Note:** If you change any of the settings in the **Create Reports** dialog box, Security Manager Administration retains these settings during the current session for both the **Create Report** and **Find Users by Entrust Properties** dialog boxes. A Search Criteria message on the **General** property page indicates that non-default criteria are in effect.

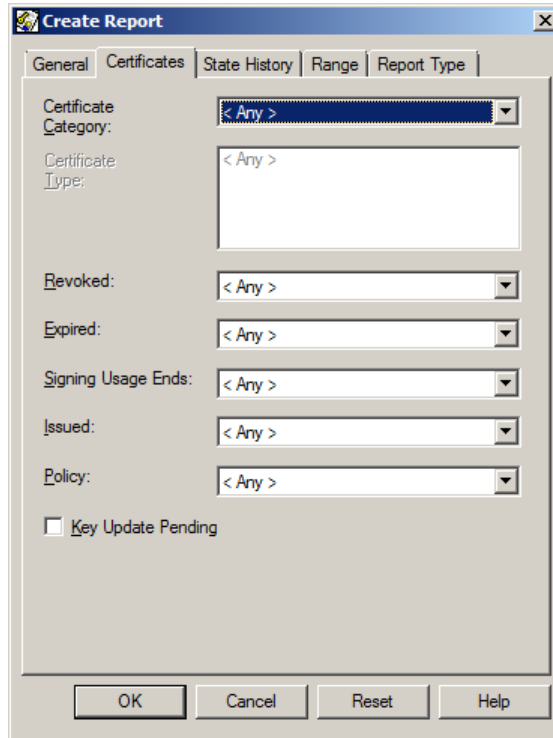
---

- 3** To revert all report options back to the defaults, click **Reset**.
- 4** Choose a property page, and define the search parameters for your report:
- 5** Click the **General** tab.
  - In the **DN** field, enter the distinguished name (DN) of the user you want to find.

When you search for entries with directory attribute values that include special characters, the values you enter must match the directory entry.

Optionally, you can use wildcards. Wildcards let you search for partial attributes. Add an asterisk (\*) with a partial search string to find users that include the search string information. For example, enter \*dr\* to find all entries named Andrew and Drew. Enter dr\* to exclude Andrew and find only entries beginning with the letters dr.
  - In the **Current State** drop-down list, select a user state to find users in a specific state, or select **<Any>** to include all user states. For details about user states, see [“User states” on page 131](#).
  - In the **Role** drop-down list, select a role to find users with a specific role, or select **<Any>** to find users with any role. For more information about roles, see [“Administering roles” on page 353](#).
  - In the **Group** drop-down list, select a group to find users who belong to that specific group, **All groups** to find users who belong to all groups, or **<Any>** to find users who belong to any group. For more information about groups, see [“Administering groups” on page 329](#).
  - In the **Attribute Certificates** drop-down list, select **<With>** to find users with attribute certificates, **<Without>** to find users without attribute certificates, or **<Any>** to find users with or without attribute certificates. For more information about attribute certificates, see [“Administering attribute certificates” on page 635](#).
  - In the **Activation Codes** drop-down list, select **<With>** to find users with activation codes, **<Expired>** to find users with expired activation codes, or **<Any>** to find users with or without activation codes. For more information about activation codes, see [“Managing activation codes” on page 153](#).

6 Click the **Certificates** tab.



The screenshot shows the 'Create Report' dialog box with the 'Certificates' tab selected. The dialog has five tabs: 'General', 'Certificates', 'State History', 'Range', and 'Report Type'. The 'Certificates' tab contains the following fields:

- Certificate Category:** A dropdown menu with '< Any >' selected.
- Certificate Type:** A text box with '< Any >' inside.
- Revoked:** A dropdown menu with '< Any >' selected.
- Expired:** A dropdown menu with '< Any >' selected.
- Signing Usage Ends:** A dropdown menu with '< Any >' selected.
- Issued:** A dropdown menu with '< Any >' selected.
- Policy:** A dropdown menu with '< Any >' selected.
- ☐ **Key Update Pending**

At the bottom of the dialog are four buttons: 'OK', 'Cancel', 'Reset', and 'Help'.

---

**Note:** Using the **Revoked**, **Expired**, **Signed Expired**, or **Issued** drop-down lists, you can specify the latest certificate as part of your search criteria. However, Security Manager can distinguish the latest certificate only if Security Manager 7.0 or later generated the certificate. Certificates created by an earlier version of Security Manager are not considered a latest certificate and are not included in the search results.

---

- In the **Certificate Category** field, select a certificate category to find users with certificates from a specific certificate category, or select **<Any>** for any certificate category.  
If you selected a certificate category, the **Certificate Type** field becomes available.
- In the **Certificate Type** field, select a certificate type to search for users with a specific certificate type, or select **<Any>** to find users with any certificate type for the certificate category.
- In the **Revoked** drop-down list, select:

- **<Any>** to find users with any certificates during the date range, regardless of revoked status
- **<All revoked>** to find users with all their certificates revoked during the date range
- **<Some revoked>** to find users with at least one revoked certificate during the date range
- **<None revoked>** to find users with no revoked certificates during the date range
- **<Latest revoked>** to find users with any of their latest certificates revoked during the date range  
The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).
- **<Latest not revoked>** to find users with none of their latest certificates revoked during the date range  
The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).
- In the **Expired** drop-down list, select:
  - **<Any>** to find users with any certificates during the date range, regardless of expired status
  - **<All expired>** to find users with all their certificates expired during the date range
  - **<Some expired>** to find users with at least one expired certificate during the date range
  - **<None expired>** to find users with no expired certificates during the date range
  - **<Latest expired>** to find users with any of their latest certificates expired during the date range  
The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).
  - **<Latest expired in current stream>** to find users with any of their latest certificates expired in the current stream during the date range  
The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).  
A stream for a user is identified by a certificate type and certificate definition. A user can have more than one stream if the user ever changed certificate types. The current stream is identified by the user's current certificate type.

- **<Latest not expired>** to find users with none of their latest certificates expired during the date range  
The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).
- In the **Signing Usage Ends** drop-down list, select:
  - **<Any>** to find users with any private signing keys, regardless of expired status
  - **<All expired>** to find users with all their private signing keys expired
  - **<Some expired>** to find users with at least one of their private signing keys expired during the date range
  - **<None expired>** to find users with no expired certificates
  - **<Latest expired>** to find users with any of their latest private signing keys expired  
The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).
  - **<Latest expired in current stream>** to find users with any of their latest private signing keys expired in the current stream during the date range  
The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).  
A stream for a user is identified by a certificate type and certificate definition. A user can have more than one stream if the user ever changed certificate types. The current stream is identified by the user's current certificate type.
  - **<Latest not expired>** to find users with none of their latest private signing keys expired  
The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).
- In the **Issued** drop-down list, select:
  - **<Any>** to find all users with any certificates during the date range, regardless of issued status
  - **<All issued>** to find all users with all their certificates issued during the date range
  - **<Some issued>** to find all users with at least one issued certificate during the date range
  - **<None issued>** to find all users with no issued certificates during the date range

- **<Latest issued>** to find all users with any of their latest certificates issued during the date range

The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).

- **<Latest not issued>** to find all users with none of their latest certificates issued during the date range

The latest certificate is the user's most recently dated certificate of any certificate definition (encryption, verification, or any other certificate definition).

---

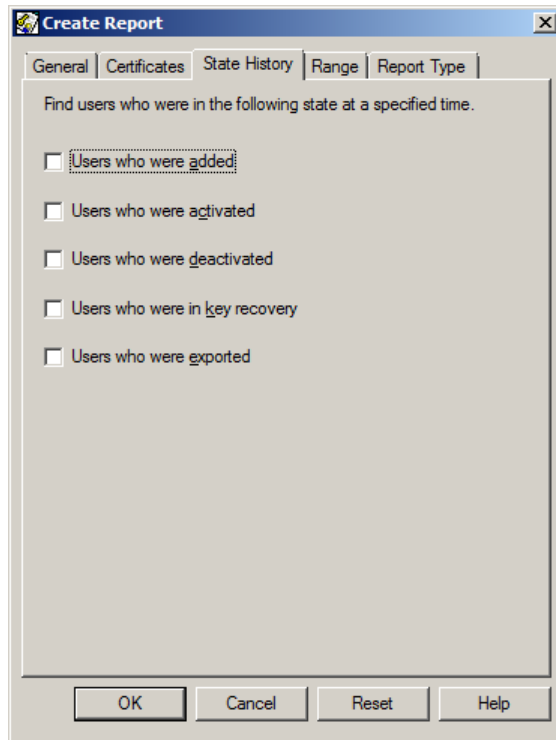
**Note:** The time the certificate is issued is not the same as the time the user is activated. The certificate issue time is set to 30 minutes after the user is activated.

---

- In the **Policy** drop-down list, select:
  - **<Default lifetime>** to find all users with policy certificates set to the default lifetime.
  - **<Specified lifetime>** to find all users with policy certificates set to a specified lifetime.
  - **<Specified expiry date>** to find all users with policy certificates set to expire on a specific date.
  - **<Any>** to find all users with policy certificates, regardless of lifetime.
- To find users with a pending key update, select **Key Update Pending**.

**7** Click the **State History** tab.





- To find users who were in the Added state during the date range, select **Users who were added**.
- To find users who were in the Active state during the date range, select **Users who were activated**.
- To find users who were in the Deactivated state during the date range, select **Users who were deactivated**.
- To find users who were in the Key Recovery state during the date range, select **Users who were in key recovery**.
- To find users who were in the Export state during the date range, select **Users who were exported**.

**8** Click the **Range** tab.

**Create Report**

General | Certificates | State History | **Range** | Report Type

Find users for all dates or specify a date and time range to limit the results.

☒ **All dates:**

☐ **Dates in Range:**

Range

From: Date: 10 - 11 - 2010 Time: 11 : 30 : 12 AM

To: 10 - 12 - 2010 11 : 30 : 12 AM

☒ Search to current date and time

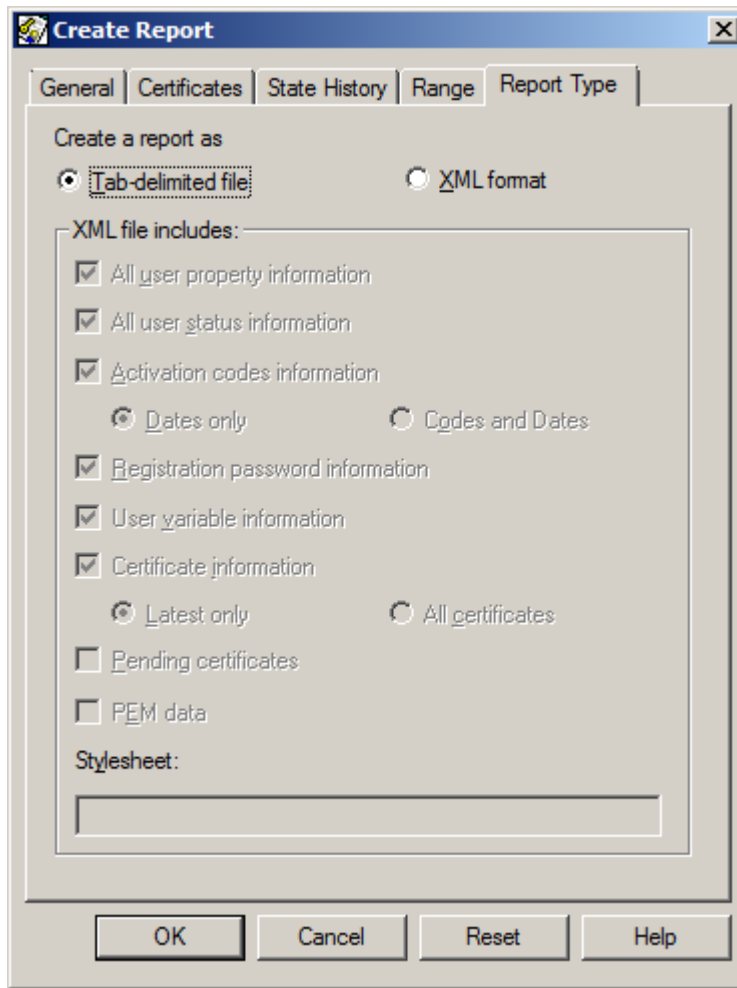
OK Cancel Reset Help

---

**Note:** You can specify dates in the future. For example, you might want to find users whose certificates will expire in the next week.

---

- To search for users , click **All dates**.
  - To specify a date range, click **Dates in Range** and then
    - Use the **From** combo boxes to enter or select a starting date and time.
    - To use the current date and time as the end of the date range, select **Search to current date and time**.
    - To specify a custom end date and time other than the current date and time, deselect **Search to current date and time** and then use the combo boxes to enter or select an ending date and time.
- 9** Click the **Report Type** tab.
- The **Report Type** property page appears.



- 10** If you want to create a tab-delimited text-format report, click **Tab-delimited file**.
- 11** If you want to create an XML-format report:
  - a** Click **XML format**.  
The options in the **XML file includes** pane become available.
  - b** In the **XML file includes** pane, use the options to select the information you want in the XML file.
  - c** If you want to use an XML stylesheet, enter its full path and file name into the **Stylesheet** field.
- 12** Click **OK**.

The **Save As** dialog box appears.

- 13** Choose a destination folder and type a name for your report.

If you are creating an XML-format report, the default file extension is `.xml`. If you are creating a tab-delimited report, the default file extension is `.txt`.

- 14** Click **Save**.

Security Manager generates the report. When it completes producing the report, a dialog box appears indicating successful completion of the operation.

- 15** Click **OK**.

You have now generated a report. Use your browser to view XML-format reports. You can view tab-delimited text files with any spreadsheet application.

---

**Attention:** Guard these reports carefully. The reports may list user setup information (reference numbers and authorization codes) that is used to create user profiles.

---

# Report fields

Table 64 lists the possible report fields as they appear in XML-format report files, and contains a brief description of each field. [Table 65 on page 632](#) lists the possible report fields as they appear in tab-delimited report files with a brief description of each field.

Individual fields in a report are blank if the corresponding characteristic is not applicable to that user. For example, a user may not have a `SubjectAltName`, or may not have any export information if they have never been exported. Similarly, a new user does not have a previous state, and there is no revocation information for a certificate that has never been revoked.

Sections of an XML-format report appear depending on the choices you selected on the **Report Type** property page. For example, if you do not check **Activation codes information**, no information in the Activation codes section of the report is found.

**Table 64:** Description of user report fields in XML-format reports

Field	Description
User DN	User's distinguished name (DN); for example, cn=Alice Gray,dc=Company One,dc=com.
User properties	
	User's alternative identification, such as an email address, a uniform resource identifier, or an IP address.
Cert category	User's certificate category—Enterprise or Web.
Cert type	User's certificate type; for example, Default, Roaming Server, or Timestamping Agent.
Admin role	User's administration role; for example, End User, Administrator, or Security Officer.
Group list	Whether the user is a member of all groups (allGroups=true or allGroups=false), followed by a list of the user's groups if the user is not a member of all groups.
Cert policy OIDs	Whether the user has the default encryption or verification OIDs (for example, encryption default=true or verification default=false), followed by a list of the OID elements if the user has non-default policy OIDs.
Cert lifetime policy	Whether the user has the default lifetime policy (default=true or false), followed by more information if the user has a non-default lifetime or expiry date assigned. If the user has a lifetime defined, the information includes the lifetime for the encryption and verification certificates and the signing key. If the user has an expiry date, the information includes the <b>use until</b> date and the <b>verify until</b> date.

**Table 64:** Description of user report fields in XML-format reports (continued)

Field	Description
Activation codes	
Reference number	Reference number from the user's activation codes. Make sure you keep this information confidential and give it to the user securely according to the security policy of your organization. Remind the user to keep the information confidential until it is used to log in to Security Manager. Depending on the security policy of your organization, you may not be able to view this information.
Authorization code	Authorization code from the user's activation codes. Make sure you keep this information confidential and that you give it to the user securely according to the security policy of your organization. Remind the user to keep the information confidential until it is used to log in to Security Manager. Depending on the security policy of your organization, you may not be able to view this information.
Create date	Date and time the Administrator created the activation codes.
Expire date	Date and time at which the activation codes expire if the user is not activated.
Registration password	This section is used by Administration Services.
User status information	
State	User's current state, which may be one of the following: Added, Activated, Deactivated, Recover, Imported (key recover), Export, Export Hold, or DN Change (target).
Last state	User's state prior to current state (see State).
Pending key update	Whether the user has a key update pending; true if the user has a key update pending; otherwise false.
Export information	Date and time at which the user was exported (exportDate), and the DN of the CA to which the user was exported (CADN).
Import CADN	The DN of the CA from which the user was imported.
Add date	Date and time the user was added.
Activate date	Date and time the user entered the activation codes and created a profile using an Entrust desktop application (for example, Entrust Entelligence, or Security Manager Administration in the case of Entrust PKI administrators).
Deactivate date	Date and time a Security Officer or Administrator deactivated the user.

**Table 64:** Description of user report fields in XML-format reports (continued)

Field	Description
Start keyrec date	Date and time a Security Officer or Administrator started a key recovery operation for the user.
Complete keyrec date	Date and time the user completed a key recovery operation in the Entrust desktop application or Security Manager Administration (in the case of Entrust PKI administrators).
Pending DN	User's current DN and user's new DN.
FCS variable values	A list of the user's <code>master.certspec</code> variables with their corresponding values.
Certificates	The following information, as applicable, is listed for each of the user's certificates.
Type	Certificate type: encryption, verification, or encryption verification (dual-purpose).
Serial number	The serial number of the certificate.
Issuer DN	The DN of the CA that issued the certificate.
Authority key ID	The encoded value of the CA's key identifier, which identifies the CA that signed the certificate.
Subject DN	The user's DN.
	A list of the user's alternative identifiers (for example, email or IP address).
Issue date	Date and time the certificate was issued.
Expire date	Date and time the certificate expires.
Private key validity	Date and time the private signing key expires (for verification certificates).
Key usage	The purpose of each key associated with the certificate: key encipherment or digital signature.
Extended key usage	A list of the extended key usage OIDs in the certificate.
CRL distribution points	A list of the DNs identifying the CDPs.
Key algorithm	The algorithm used for the signing private key (for example, 1024-bit DSA).

**Table 64:** Description of user report fields in XML-format reports (continued)

Field	Description
Private key archived	Whether the private signing key is backed up. ( <code>privateKeyArchived=true</code> or <code>false</code> ).
Certificate published	Whether the certificate is published to the directory ( <code>published=true</code> or <code>false</code> ).
Entrust Authority version	Security Manager version that issued the certificate.
Certificate policy OIDs	A list of the policy OIDs in the certificate.
Revocation information	<p>Revocation information:</p> <ul style="list-style-type: none"> <li>the date and time the certificate was revoked (<code>revokeDate</code>)</li> <li>the reason for revocation (<code>revokeReason</code>), which may be On Hold, Superseded, Key Compromise, Affiliation Change, Unspecified, or Cessation of Operation</li> <li>a revocation comment supplied by the Administrator or Security Officer (<code>revokeComment</code>)</li> <li>the date of compromise if the reason for revocation is key compromise (<code>compromiseDate</code>), which is the date when the certificate was last known to be uncompromised</li> </ul> <p>See the <i>Security Manager Administration User Guide</i> for information about revocation reasons.</p>

**Table 65:** Description of user report fields in tab-delimited reports

Field	Description
Name	User's distinguished name (DN); for example, <code>cn=Alice Gray,dc=Company One,dc=com</code> .
State	User's current state, which may be one of the following: Added, Activated, Deactivated, Revoked, Recover, Imported (key recover), Export, Export Hold, Expired, or DN Change (target).
Updated by	The last user who performed an encryption key update. The value is one of Client, Administrator, or Unknown.



**Table 65:** Description of user report fields in tab-delimited reports (continued)

Field	Description
Reference #	Reference number from the user's activation codes. Make sure you keep this information confidential and give it to the user securely according to the security policy of your organization. Remind the user to keep the information confidential until it is used to log in to Security Manager. Depending on the security policy of your organization, you may not be able to view this information.
Activation	Authorization code from the user's activation codes. Make sure you keep this information confidential and that you give it to the user securely according to the security policy of your organization. Remind the user to keep the information confidential until it is used to log in to Security Manager. Depending on the security policy of your organization you may not be able to view this information.
Added	Date and time the user was added.
Activated	Date and time the user entered the activation codes and created a profile using an Entrust desktop application (for example, Entrust Entelligence or Security Manager Administration in the case of Entrust PKI administrators).
Key Rec Init	Date and time a Security Officer or Administrator started a key recovery operation for the user.
Key Rec Done	Date and time the user completed a key recovery operation in the Entrust desktop application or Security Manager Administration (in the case of Entrust PKI administrators).
User Deactivated	Date and time a Security Officer or Administrator deactivated the user.
Deactivated Reason	Reason for deactivating the user. Entered automatically by Security Manager Administration. For example, "User deactivated by Administrator."
Previous status	User state prior to current state (see State).
Enc Updt Pend	Whether or not encryption key pair update is pending for a user.
Signing Cert Desc	Describes the type of signing key pair (for example, 1024-bit DSA).
Signing Cert Issue	Date and time the user's current verification certificate was issued.
Signing Cert Expiry	Date and time the user's current verification certificate expires.
Signing Cert Revoked	Date and time the user's most recent verification certificate was revoked.
Sign. Cert Revok. Reason	Reason for revoking the user's verification certificate. May be one of On Hold, Superseded, Key Compromise, Affiliation Change, Unspecified, or Cessation of Operation. See <a href="#">"Revoking user certificates" on page 174</a> for information about revocation reasons.

**Table 65:** Description of user report fields in tab-delimited reports (continued)

Field	Description
Encrypt Cert Desc	Describes the type of encryption key pair (for example, 1024-bit RSA).
Encrypt Cert Issue	Date and time the user's current encryption certificate was issued.
Encrypt Cert Expiry	Date and time the user's current encryption certificate expires.
Encrypt Cert Revoked	Date and time the user's most recent encryption certificate was revoked.
Encr. Cert Revok. Reason	Reason for revoking the user's encryption certificate. May be one of On Hold, Superseded, Key Compromise, Affiliation Change, Unspecified, or Cessation of Operation. See <a href="#">"Revoking user certificates" on page 174</a> for information about revocation reasons.
Pending DN	User's current distinguished name and user's new distinguished name.

# Administering attribute certificates

Security Manager allows you to issue attribute certificates directly to individual users. Attribute certificates are a mechanism, based on the X.509 standard, used to convey privilege information to support authorization services. The specific privilege information that you convey could be the assignment of a user to a role or assignment of specific privileges to a user or role.

To administer attribute certificates, your role must include permissions to create and modify user policies. For information about roles and policies, see [“Administering roles” on page 353](#).

---

**Note:** If you plan to issue attribute certificates to end users, you should update your directory schema to include the `attributeCertificateAttribute` attribute and the `pmiUser` object class (the default attribute and object class used for issuing attribute certificates). For information on Entrust schema requirements, see the white paper *Security Manager Directory Schema Requirements*. Check the Entrust TrustedCare Online Web site (<https://secure.entrust.com/trustedcare>) for a listing of related white papers and where to find them.

---

This chapter includes the following sections:

- [“About attribute certificates” on page 636](#)
- [“Steps for working with attribute certificates” on page 638](#)
- [“Creating custom attribute certificate types” on page 639](#)
- [“Issuing attribute certificates” on page 640](#)
- [“Managing attribute certificates” on page 643](#)

# About attribute certificates

Attribute certificates are a mechanism, based on the X.509 standard, used to convey privilege information to support authorization services. The specific privilege information that you convey could be the assignment of a user to a role or assignment of specific privileges to a user or role.

Attribute certificates are similar to public-key certificates in that they are digitally signed by an issuing authority and they provide a secure binding of the certificate content to the entity to which the certificate is issued. In the case of public-key certificates, the issuing authority is a Certification Authority (CA) and the certificate binds a public key to the entity. In the case of attribute certificates, the issuing authority is an Attribute Authority (AA) and the certificate binds a set of attributes or privileges to the entity. Attribute certificates do not contain public keys.

---

**Note:** A single issuing authority can act as both a CA and an AA, issuing public key and attribute certificates to its user population.

---

Topics in this section:

- [“How are attribute certificates used?” on page 636](#)
- [“Examples of using attribute certificates” on page 637](#)
- [“Attribute certificates and Security Toolkit for the Java Platform” on page 637](#)
- [“About Security Toolkit for Java Platform” on page 637](#)

## How are attribute certificates used?

You use attribute certificates for a number of purposes, including authorization of users to perform specific functions or assignment of users to roles.

Security Manager now supports the issuance of general attribute certificates to users. You can customize the privilege information contained in attribute certificates to meet specific application and operating environment requirements. You use the same tools to define the attributes that carry privilege information as the tools you use to customize public-key certificates. Once issued, attribute certificates are stored in the user's directory entry and are retrievable by applications built with Entrust Authority Security Toolkit for Java Platform.

## Examples of using attribute certificates

The following examples show how to use attribute certificates to convey permissions and ensure authentication.

A hospital can issue attribute certificates to its emergency room medical staff, conveying permissions to prescribe certain drugs. Security Toolkit for Java Platform is used to develop an application that determines whether a particular staff member can dispense a particular medication. The Java application calls the Security Toolkit for Java Platform to obtain the staff member's attribute certificate in the directory, validate it, and then check the list of drugs in the attribute certificate.

In this way, the Java application, built with Security Toolkit for Java Platform, checks the attribute certificate to verify that the user has valid permissions.

## Attribute certificates and Security Toolkit for the Java Platform

Security Toolkit for Java Platform enables developers to build applications that work with attribute certificates. Specifically, Security Toolkit for Java Platform

- retrieves the user's attribute certificates from the directory
- ensures that the signature on the attribute certificate is from a valid source
- validates the attribute certificate's integrity
- passes the contents of the attribute certificate to the application

## About Security Toolkit for Java Platform

Security Toolkit for Java Platform enables developers to build applications with flexible, modular security services for encryption, decryption, digital signatures, and verification with X.509 managed certificates. Security Toolkit for Java Platform makes it easy to build interoperable Java applications, applets, or servlets with multiple PKI solutions. For more information, visit [www.entrust.com](http://www.entrust.com) and search on Security Toolkit for Java Platform.

# Steps for working with attribute certificates

The steps involved in working with attribute certificates are as follows:

- 1** Create a new attribute certificate type by editing the `master.certspec` file. See [“Creating custom attribute certificate types” on page 639](#).
- 2** Find the user you want to issue an attribute certificate to. See [“Finding users” on page 134](#).
- 3** Issue an attribute certificate to the user. See [“Issuing attribute certificates” on page 640](#).
- 4** Now you can save, reissue, or delete the certificate, as well as view and modify its properties. See [“Managing attribute certificates” on page 643](#).

---

**Note:** You can issue attribute certificates only to active users.

---

# Creating custom attribute certificate types

In order to issue an attribute certificate to a user, you must first create a custom attribute certificate type by editing the `master.certspec` file. You create a custom certificate type using a certificate type description and one or more certificate attributes.

For complete details on creating a custom attribute certificate type, see [“Customizing policy certificates” on page 571](#). Although [“Customizing policy certificates”](#) deals with policy certificates, which are one form of attribute certificates, the section covers all of the steps necessary for creating a new attribute certificate type, including

- adding a certificate type description
- editing certificate attributes
- defining variables for a certificate attribute

After you create a custom certificate type and process the `master.certspec` file, the **Add Attribute Certificate** dialog box lists the custom certificate type the next time you open it.

# Issuing attribute certificates

After you create a custom attribute certificate type and find users, you can issue attribute certificates. For details on creating custom attribute certificate types, see [“Creating custom attribute certificate types” on page 639](#).

---

**Note:** You can issue attribute certificates only to users in the active state. However, you can manage existing attribute certificates for users in other states.

---

## To issue an attribute certificate

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Find the user whose requires an attribute certificate (see [“Finding users” on page 134](#)).
- 3 Select **Users > Selected User > Add Attribute Certificate**.

The **Add Attribute Certificate** dialog box appears.

The screenshot shows the 'Add Attribute Certificate' dialog box. It contains the following fields and options:

- Label:** A text input field.
- Lifetime (in hours):** A text input field.
- Start time:** A text input field showing '01/30/2003 01:11:38 PM'.
- ☐ Automatically reissue the certificate when it expires
- Type:** A dropdown menu showing 'Entertainment Download' with a description '(Download Access for Entertainment)'. Below it, another dropdown shows 'Live Entertainment' with a description '(Access to Live Concerts)'.
- Attributes:** A section containing:
  - Download types:** A text input field.
  - Spending limit:** A text input field.
- Download types:** A section containing:
  - A text area with the text: 'Entertainment formats that user is allowed to download. Enter a value of type: Text String Permitted values: Music Video Games'.
- Store the certificate in:** A section with two radio buttons: 'File' and 'Directory'. 'Directory' is selected.
- Buttons:** 'OK', 'Cancel', and 'Help' buttons at the bottom right.

- 4 In the **Label** field, type a name for the attribute certificate.



The name should describe the particular attribute certificate you plan to issue for this user. A user cannot have two attribute certificates with the same label.

- 5** In the **Lifetime** field, type the number of hours that you want the certificate to remain valid.

The minimum value is one hour, and the lifetime of the certificate (that is, start time plus lifetime) cannot exceed the lifetime of the CA certificate (since the CA signs the attribute certificate).

- 6** In the **Start time** field, type the time when this attribute certificate becomes active. The current time appears by default.
- 7** Select **Automatically reissue the certificate when it expires** if you want Security Manager to automatically reissue the certificate.

For details on automatic reissue of attribute certificates, see [“Automatic reissue of attribute certificates” on page 643](#).

- 8** Select one of your custom attribute certificate types in the **Type** box. When you click a certificate type, its attributes display.
- 9** Fill in the **Attributes** fields as required.

When you click in one of the fields, a description and permitted values appear in an information box below the attributes.

---

**Note:** The **Attributes** fields are case-sensitive. Type permitted values exactly as they appear in the information box.

---

- 10** Specify whether the certificate is stored in a text file or published to the directory. To store the certificate in a text file, click **File**. To store the certificate in the directory, click **Directory**.

Normally, you should store a new attribute certificate in the directory, but if you want to save it to file for future reference instead, you can do so here. You can also save an attribute certificate to file at any time after it is created. For more details on saving a certificate to file, see [“Saving attribute certificates to file” on page 648](#).

As long as you specify to store an attribute certificate in the directory, the attribute certificate is always immediately re-posted to the directory if it is reissued, replaced, or modified. However, if you specify to save the attribute certificate to file (at creation or at a later time), the certificate is not automatically re-saved to file if it is reissued, replaced, or modified, nor is it posted to the directory.

---

**Note:** If you specify to store an attribute certificate that was previously saved to file in the directory, the attribute certificate is immediately posted to the directory and reissued. The start time of the certificate does not change.

---

**11** Click **OK**.

**12** If you chose to save the attribute certificate to a file:

The **Attribute Certificate Save As** dialog box appears.

- a** Choose a destination folder and type a name for your attribute certificate file. An extension of `.txt` appends automatically to the filename.
- b** Choose either **Binary** or **PEM encoded** by clicking the appropriate option.
- c** Click **Save**.

**13** If prompted to authorize the operation, authorize the operation. See [“Authorizing sensitive operations” on page 52](#).

A dialog box informs you that the attribute certificate is created.

**14** Click **OK**.

You have now issued an attribute certificate to a user.

# Managing attribute certificates

Once you have issued an attribute certificate to a user's entry, you can reissue, delete, or replace it, as well as view its contents and save it to file. This section also includes notes on attribute certificates related to administering users.

This section shows you how to

- ["Reissuing attribute certificates" on page 643](#)
- ["Deleting attribute certificates" on page 645](#)
- ["Replacing attribute certificates" on page 647](#)
- ["Viewing the contents of attribute certificates" on page 648](#)
- ["Saving attribute certificates to file" on page 648](#)
- ["Administering users with attribute certificates" on page 650](#)

## Reissuing attribute certificates

If an attribute certificate has expired, you may want to reissue it. You can reissue attribute certificates only for users in the Active state. If you try to reissue an attribute certificate to a user in any other user state, a dialog box appears with the error message "User is not in the correct state."

This section explains how to reissue an attribute certificate to a user's entry.

### Automatic reissue of attribute certificates

To prevent expiry of attribute certificates, you can specify automatic reissue when you create, modify, or replace the certificate. For details on how to specify automatic reissue when you create an attribute certificate, see ["Issuing attribute certificates" on page 640](#). For details on how to specify automatic reissue for an existing attribute certificate, see ["Replacing attribute certificates" on page 647](#).

By default, Security Manager reissues certificates when they are within 24 hours of expiring. A periodic process checks for certificates that are about to expire. By default, the periodic process checks when Security Manager is started and every 20 hours thereafter.

To ensure that the periodic process does not spend all of its time reissuing attribute certificates, it reissues the certificates in blocks. After each block, it performs other periodic functions (such as reissuing CRLs) before reissuing the next block of attribute certificates. By default, the block size is 100 certificates.

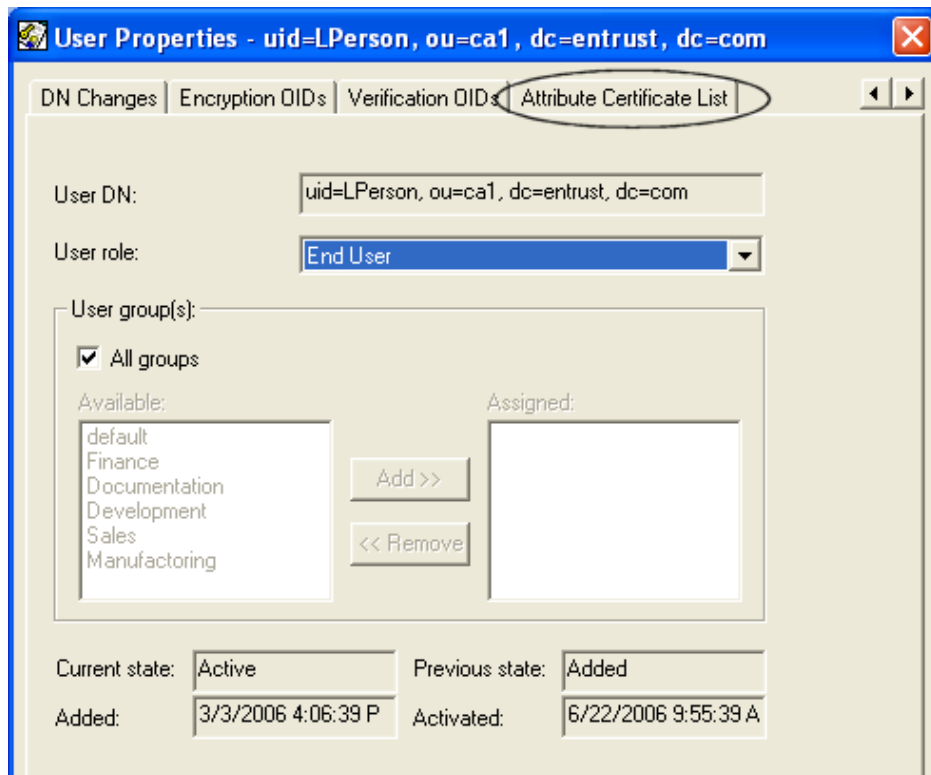
To adjust the timer controls for automatic reissue of attribute certificates, you can specify settings in the `entmgr.ini` file. For details on these settings, see the *Security Manager Operations Guide*.

If you have specified to post the attribute certificate to the directory, Security Manager also posts the certificate to the directory when Security Manager reissues

the certificate. If you have specified to save to file the attribute certificate, the certificate is not automatically re-saved to file upon reissue.

### To manually reissue an attribute certificate to a user's entry

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Find the user for whom you want to reissue an attribute certificate (see [“Finding users” on page 134](#)).
- 3 Select **Users > Selected User > Properties**.  
The **User Properties** dialog box for the selected user appears.
- 4 Scroll to and click the **Attribute Certificate List** property page.



---

**Note:** To see this property page, you must have issued an attribute certificate to the user.

---

The attribute certificates for the selected user display.

- 5 Right-click the expired certificate and click **Reissue** in the pop-up menu.
- 6 Depending on the policies and procedures of your company, one or more Authorization Required dialog boxes may appear. Authorize the transaction if required, and click **OK**.

The **Valid From** column for the reissued certificate now displays a new valid-from time. This value may be 30 minutes earlier than the time it was actually reissued. This is to allow for possible time differences between the CA and client machine; otherwise, the client might receive a certificate that is not valid until a future time.

You have now reissued an attribute certificate to a user's entry.

## Deleting attribute certificates

You may want to delete an attribute certificate. When you delete an attribute certificate, you remove it from Security Manager and from the directory, if the attribute certificate is published to the directory. However, the deleted attribute certificate is not revoked. If the attribute certificate was originally saved to file, the file is not deleted when you delete the attribute certificate.

---

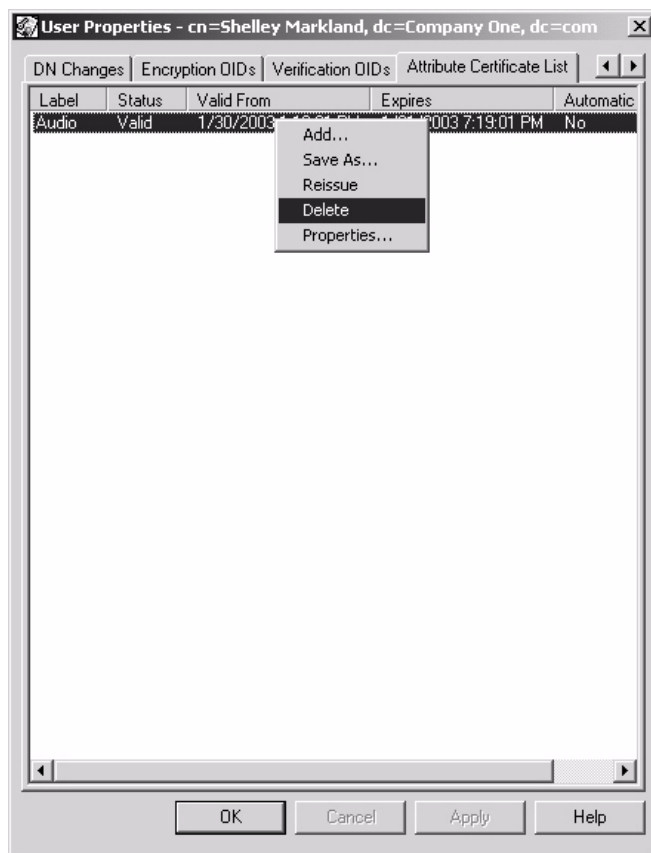
**Note:** It is not possible to revoke an attribute certificate. Since attribute certificates are designed to be short-lived certificates that you can set to renew automatically, there is no need to publish revocation lists.

---

You can delete attribute certificates from both Active and non-active users' entries.

### To delete an attribute certificate from a user's entry

- 1 Log in to Security Manager Administration (see ["Logging in to Security Manager Administration" on page 46](#)).
- 2 Find the user for whom you want to delete an attribute certificate (see ["Finding users" on page 134](#)).
- 3 Select **Users > Selected User > Properties**.  
The **User Properties** dialog box for the selected user appears.
- 4 Scroll to and click the **Attribute Certificate List** property page.  
The attribute certificates for the selected user display.
- 5 Right-click the certificate you wish to delete and select **Delete** in the pop-up menu.



- 6 Depending on the policies and procedures of your company, one or more **Authorization Required** dialog boxes may appear. Authorize the transaction if required, and click **OK**.

The certificate disappears from the certificate list.

You have now deleted an attribute certificate from a user's entry.

## Replacing attribute certificates

Once you have issued an attribute certificate to a user's entry, you can replace it with a newly issued certificate with different properties. You can replace attribute certificates only for users in the Active state. If you try to replace an attribute certificate for a user in any other state, such as Added, Deactivated, or Key Recovery, a dialog box appears with the error message "User is not in the correct state."

### To replace an attribute certificate

- 1** Log in to Security Manager Administration (see ["Logging in to Security Manager Administration" on page 46](#)).
  - 2** Find the user for whom you want to replace an attribute certificate (see ["Finding users" on page 134](#)).
  - 3** Select **Users > Selected User > Properties**.  
The **User Properties** dialog box for the selected user appears.
  - 4** Scroll to and click the **Attribute Certificate List** property page.  
The attribute certificates for the selected user display.
  - 5** Right-click the certificate you want to replace, and click **Properties** in the pop-up menu.
  - 6** When the **Attribute Certificate Properties** dialog box appears, make sure the **Attribute Certificate Information** property page is selected.
  - 7** Edit the fields you want to change. For information on the fields, see ["Issuing attribute certificates" on page 640](#).
  - 8** When you have finished editing the fields, click **OK**.
  - 9** Depending on the policies and procedures of your company, one or more **Authorization Required** dialog boxes may appear. Authorize the transaction if required, and click **OK**.
  - 10** A dialog box appears to indicate successful completion of the operation. Click **OK**.
  - 11** Click **OK** again to exit the **User Properties** dialog box.
- You have now replaced a user's attribute certificate.

## Viewing the contents of attribute certificates

After creating an attribute certificate for a user, you can view the certificate's contents. You can view the contents of attribute certificates for both Active and non-active users' entries.

### To view the contents of an attribute certificate

- 1 Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2 Find the user for whom you want to view an attribute certificate (see [“Finding users” on page 134](#)).  
The **User Properties** dialog box for the selected user appears.
- 3 Scroll to and click the **Attribute Certificate List** property page.  
The attribute certificates for the selected user display.
- 4 Right-click the certificate you wish to view and click **Properties** in the pop-up menu.
- 5 When the **Attribute Certificate Properties** dialog box appears, click the **Certificate Contents** property page.  
The certificate contents display.

---

**Note:** If you have just modified the certificate in the **Attribute Certificate Information** property page, the changes do not appear in the **Certificate Contents** property page unless you clicked **OK** after making the changes.

---

- 6 Scroll down to view the certificate's extensions and attributes.  
You have now viewed the contents of an attribute certificate.

## Saving attribute certificates to file

By default, attribute certificates are stored in the directory. You may want to save an attribute certificate to a file, rather than to the directory, so that it is available for use in applications that do not use the directory. Security Manager allows you to save attribute certificates to file in either binary or PEM-encoded text.

You can save attribute certificates to file

- when they are created, using the **Add Attribute Certificate** dialog box
- at any time after they are issued, using the **Attribute Certificate List**

This procedure assumes that you are saving an attribute certificate to a file after the certificate was created. To learn how to save an attribute certificate to a file when creating the certificate, see [“Issuing attribute certificates” on page 640](#).



You can save attribute certificates to file from both Active and non-active users' entries.

For details on the types of applications that can work with attribute certificates, see [“About attribute certificates” on page 636](#).

### To save an attribute certificate to a file

- 1** Log in to Security Manager Administration (see [“Logging in to Security Manager Administration” on page 46](#)).
- 2** Find the user whose attribute certificate you want to save to file (see [“Finding users” on page 134](#)).
- 3** Select **Users > Selected User > Properties**.  
The **User Properties** dialog box for the selected user appears.
- 4** Scroll to and click the **Attribute Certificate List** property page.  
The attribute certificates for the selected user display.
- 5** Right-click the certificate you wish to save to file and click **Save As** in the pop-up menu.

---

**Note:** You can also save an attribute certificate to file using the **Attribute Certificate Properties** dialog box. Double-click the attribute certificate and click **File** under **Store the certificate in**. This brings you to the **Attribute Certificate Save As** dialog box.

---

The **Attribute Certificate Save As** dialog box appears.

- 6** Choose a destination folder and type a name for your attribute certificate file. An extension of `.txt` appends automatically the filename.
- 7** Choose either **Binary** or **PEM encoded** by clicking the appropriate option.
- 8** Click **Save**.
- 9** Depending on the policies and procedures of your company, one or more **Authorization Required** dialog boxes may appear. Authorize the transaction if required, and click **OK**.
- 10** The **Operation Completed Successfully** dialog box appears. Click **OK**.

You have now saved a user's attribute certificate to a file.

## Administering users with attribute certificates

When administering users who have attribute certificates, keep in mind the following:

- When a user is archived, the user's attribute certificates are not archived.
- When a user is exported, the user's attribute certificates are not exported.
- When a user's DN is changed, the user's attribute certificates are reissued (with the new DN).
- When a user's public-key certificates are revoked, the user's attribute certificates are not revoked. Since attribute certificates are designed to be short-lived certificates that you can set to renew automatically, there is no need to publish revocation lists.

## Bulk commands reference

This appendix provides the syntax for bulk commands. For information about creating and processing bulk files, see [“Performing bulk operations” on page 275](#).

This appendix contains the following sections:

- [“Authorizing operations in bulk files” on page 652](#)
- [“Group bulk commands” on page 653](#)
- [“Searchbase bulk commands” on page 655](#)
- [“User bulk commands” on page 658](#)

# Authorizing operations in bulk files

Most bulk commands require authorization before Security Manager Administration can process the command. The role of the Entrust PKI administrator who is processing the bulk file determines which commands require authorization, and the number of authorizations required (see [“Administering roles” on page 353](#)).

If a command in your bulk file requires authorization, processing pauses and Security Manager Administration prompts you to authorize the operation. After authorization the operation, Security Manager Administration resumes processing. Security Manager does not require you to authorize subsequent occurrences of the same command.

If you are prompted to provide authorization for a command and you cancel the authorization, Security Manager Administration stops processing the bulk file.

To avoid having to wait to enter authorizations for commands that Security Manager Administration might not process until late into the bulk processing operation, you can collect authorization at the start of the bulk file using the `authorize` command.

## Syntax

```
authorize <command>
```

Where `<command>` specifies the command you want to authorize before Security Manager Administration processes the command.

For example, `authorize user_add` causes Security Manager Administration to prompt you to authorize all instances of the `user_add` command.

If you include the `authorize` command for a bulk command that does not require authorization Security Manager Administration does not prompt you to provide authorization for that command.

# Group bulk commands

Using bulk operations, you can create and delete groups, and remove all users from groups. For more information about administering groups, see [“Administering groups” on page 329](#).

This section describes the following group bulk commands:

- [“group” on page 653](#)
- [“group\\_create” on page 653](#)
- [“group\\_delete” on page 654](#)
- [“group\\_removeallusers” on page 654](#)

## group

The `group` command identifies a group object.

In bulk operations, you identify a group by specifying the name of the group. This is done by declaring a group object. A group object is used in a bulk script for manipulating a group. When you want to perform an operation on a group (for example, delete it), you declare a group object and associate that group object with the name of the group you wish to manipulate. When you declare a group object, you also associate it with a group ID, which provides a handle for referencing the group object later in the script.

### Syntax

```
group <groupID> <groupname>
```

Where:

- `<groupID>` is the identifier of the group object (group ID).
- `<groupname>` is the name for the group.

## group\_create

The `group_create` command creates a new group. For more information about creating groups, see [“Creating groups” on page 332](#).

### Syntax

```
group_create <groupID>
```

Where `<groupID>` is the identifier of the group object. The group ID was specified in the `group` command (see [“group” on page 653](#)).

## group\_delete

The `group_delete` command deletes a group. Before you delete a group, you must remove all members from the group with the `group_removeallusers` command (see [“group\\_removeallusers” on page 654](#)). As all members must belong to at least one group, one group must always exist. When Security Manager is installed, it creates a default group (called Default) which can you can later delete as you create new groups.

For more information about deleting groups, see [“Deleting groups” on page 339](#).

### Syntax

```
group_delete <groupID>
```

Where `<groupID>` is the identifier of the group object. The group ID was specified in the `group` command (see [“group” on page 653](#)).

## group\_removeallusers

The `group_removeallusers` command removes all users from a group. All users must belong to at least one group. You cannot remove all users from a group if at least one of those users does not belong to another group.

For more information about removing users from groups, see [“Removing members from groups” on page 336](#).

### Syntax

```
group_removeallusers <groupID>
```

Where `<groupID>` is the identifier of the group object. The group ID was specified in the `group` command (see [“group” on page 653](#)).

# Searchbase bulk commands

Using bulk operations, you can create, delete, and modify searchbases. For more information about administering searchbases, see [“Administering searchbases” on page 341](#).

This section describes the following searchbase bulk commands:

- [“searchbase” on page 655](#)
- [“searchbase\\_add” on page 656](#)
- [“searchbase\\_delete” on page 656](#)
- [“searchbase\\_setdn” on page 656](#)
- [“searchbase\\_setlabel” on page 657](#)
- [“searchbase\\_userscansearch” on page 657](#)

## searchbase

In bulk operations, you identify a searchbase by specifying its distinguished name (DN). This is done by declaring a searchbase object. A searchbase object is used in a bulk script for manipulating a searchbase. When you want to perform an operation on a searchbase (for example, change its label), you declare a searchbase object and associate that searchbase object with the DN of the searchbase you wish to manipulate. When you declare a searchbase object, you also associate it with a searchbase ID, which provides a handle for referencing the searchbase object later in the script.

### Syntax

```
searchbase <searchbaseID> <searchbaseLabel> <searchbaseDN>  
<usersCanSearch>
```

Where:

- `<searchbaseID>` is the identifier of the searchbase object (searchbase ID).
- `<searchbaseLabel>` is the name of the searchbase. This is the name that users see in interfaces such as Security Manager Administration.
- `<searchbaseDN>` is the DN of the searchbase.
- `<usersCanSearch>` specifies whether all users are allowed to search for recipients in this searchbase (this argument has a value of either `TRUE` or `FALSE`). Enter `FALSE` in the rare circumstance when you do not want users to encrypt files for other users in the searchbase.

## searchbase\_add

The `searchbase_add` command adds a new searchbase to Security Manager. Before you add a searchbase with bulk commands, you must add a searchbase to the Security Manager directory (see [“Adding searchbases to the directory” on page 345](#)).

For more information about adding searchbases, see [“Adding searchbases” on page 344](#).

### Syntax

```
searchbase_add <searchbaseID>
```

Where `<searchbaseID>` is the identifier of the searchbase object. The searchbase ID was specified in the `searchbase` command (see [“searchbase” on page 655](#)).

## searchbase\_delete

The `searchbase_delete` command deletes a searchbase. For more information about deleting searchbases, see [“Deleting searchbases” on page 351](#).

When you delete a searchbase, the associated directory entry is not automatically deleted. Users belonging to the searchbase are not deleted from the directory. Rather, users' association to the deleted searchbase is removed.

### Syntax

```
searchbase_delete searchbaseID
```

Where `<searchbaseID>` is the identifier of the searchbase object. The searchbase ID was specified in the `searchbase` command (see [“searchbase” on page 655](#)).

## searchbase\_setdn

The `searchbase_setDN` command specifies the distinguished name (DN) of a searchbase. For more information about setting the distinguished name of a searchbase, see [“Modifying searchbases” on page 349](#).

To include special characters such as a comma or backslash in the DN, you must precede the special character with a backslash. For more information, see [“Using special characters in user names” on page 133](#).

### Syntax

```
searchbase_setdn <searchbaseID> <searchbaseDN>
```

Where:

- `<searchbaseID>` is the identifier of the searchbase object. The searchbase ID was specified in the `searchbase` command (see [“searchbase” on page 655](#)).



- `<searchbaseDN>` is the DN of the searchbase.

## searchbase\_setlabel

The `searchbase_setlabel` command specifies the name of a searchbase. This is the name that users see in interfaces such as Security Manager Administration.

### Syntax

```
searchbase_setlabel <searchbaseID> <searchbaselabel>
```

Where:

- `<searchbaseID>` is the identifier of the searchbase object. The searchbase ID was specified in the `searchbase` command (see [“searchbase” on page 655](#)).
- `<searchbaselabel>` specifies the name of the searchbase. This is the name that users see in interfaces such as Security Manager Administration.

## searchbase\_userscansearch

The `searchbase_userscansearch` command specifies whether the searchbase is searchable by users. You should only set this command to `FALSE` in the rare circumstance where you do not want users to encrypt files for other users in the searchbase.

### Syntax

```
searchbase_userscansearch <searchbaseID> TRUE|FALSE
```

Where:

- `<searchbaseID>` is the identifier of the searchbase object. The searchbase ID was specified in the `searchbase` command (see [“searchbase” on page 655](#)).
- `TRUE` specifies that users can search this searchbase. `FALSE` specifies that users are not able to search this searchbase.

# User bulk commands

This section describes the bulk commands for administering users in bulk. For more information about administering users, see [“Administering users” on page 129](#).

To perform bulk operations, you write scripts and save them in a bulk file. You then process the files in the Bulk Console window (see [“Performing bulk operations” on page 275](#)). To make writing bulk scripts easier, you can search for users and save the results to a text file. For information about finding users, see [“Finding users” on page 134](#).

This section contains the following topics:

- [“user” on page 659](#)
- [“user\\_add” on page 659](#)
- [“user\\_addtodn” on page 660](#)
- [“user\\_applyproperties” on page 660](#)
- [“user\\_archive” on page 660](#)
- [“user\\_assigndn” on page 661](#)
- [“user\\_cancelchangedn” on page 661](#)
- [“user\\_cancelhold” on page 661](#)
- [“user\\_cancelrecover” on page 662](#)
- [“user\\_convertv1” on page 662](#)
- [“user\\_createdirent” on page 662](#)
- [“user\\_deactivate” on page 663](#)
- [“user\\_deletedirent” on page 663](#)
- [“user\\_export” on page 664](#)
- [“user\\_free” on page 666](#)
- [“user\\_import” on page 666](#)
- [“user\\_notifyclient” on page 666](#)
- [“user\\_reactivate” on page 667](#)
- [“user\\_recover” on page 667](#)
- [“user\\_reissueactivationcodes” on page 667](#)
- [“user\\_renamedirent” on page 668](#)
- [“user\\_restoretodir” on page 668](#)
- [“user\\_retrieve” on page 668](#)
- [“user\\_revoke” on page 669](#)
- [“user\\_setattribute” on page 670](#)

- [“user\\_setparentdn” on page 670](#)
- [“user\\_setproperty” on page 671](#)
- [“user\\_settemplate” on page 677](#)
- [“user\\_updatekeypairs” on page 677](#)
- [“user\\_refresh\\_subaltname\\_from\\_dir” on page 678](#)

## user

In bulk operations, you identify a user by specifying the distinguished name (DN). This is done by declaring a user object. A user object is used in a bulk script for manipulating a user. When you want to perform an operation on a user (for example, recover a user), you declare a user object and associate that user object with the DN of the user you wish to manipulate. When you declare a user object, you also associate it with a user ID, which provides a handle for referencing the user object later in the script.

### Syntax

```
user <userID> [<userDN>] [New]
```

Where:

- `<userID>` is the identifier of the user object (user ID).
- `<userDN>` is the complete DN of a user. The DN is optional. For example, when creating a new directory entry with a template, you cannot specify the DN because the user's DN does not yet exist.
- `New` is an optional parameter to indicate that you are creating a new, customized directory entry, as opposed to a template-based user.

## user\_add

The `user_add` command adds a user to Security Manager. The user must already exist in the directory but not be already added to Security Manager. The `user_add` command also commits any user properties that are set with the `user_setproperty` command.

If the Entrust PKI administrator processing the bulk script has permission to view activation codes, then the activation codes appear in the bulk output log file. You must distribute these activation codes to your users so they can create digital IDs (see [“Distributing activation codes” on page 153](#)).

### Syntax

```
user_add <userID>
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).

## user\_addtodn

The `user_addtodn` command specifies an additional attribute to include in the user's distinguished name (DN) when the user's directory entry is created. This command applies only when creating directory entries using a template.

The attribute specified must be identified in the template as optional for the DN. If the attribute is identified as not allowed in the DN, or is not identified at all in the template, then an error is returned.

Specifying an attribute already in the DN has no effect, though no error is returned.

### Syntax

```
user_addtodn <userID> <attribute>
```

Where:

- <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).
- <attribute> is the name of the attribute to include in the DN.

## user\_applyproperties

The `user_applyproperties` command applies to the user ID any user properties that are modified using the `user_setproperty` command. Security Manager Administration does not apply changes to user properties until it processes the `user_applyproperties` command.

### Syntax

```
user_applyproperties <userID>
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).

## user\_archive

The `user_archive` command archives a user's key history. When a user is archived, the user's key history is written to an archive file on the server hosting Security Manager. The Security Manager database is cleared of all information about the user.

An archived user's key history is retrievable from the archive file using the `user_retrieve` command (see [“user\\_retrieve” on page 668](#)).

## Syntax

```
user_archive <userID>
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).

## user\_assigndn

The `user_assigndn` command assigns a new distinguished name (DN) to a user. The new DN must already exist in the directory.

## Syntax

```
user_assigndn <userID> <newDN>
```

Where:

- <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).
- <newDN> is the user’s new DN.

## user\_cancelchangedn

The `user_cancelchangedn` command cancels a change DN operation previously performed on an Entrust PKI user. All references to the new DN are removed from the Security Manager database. This command does not work if the user has already logged in and completed the change DN operation.

## Syntax

```
user_cancelchangedn <userID>
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).

## user\_cancelhold

The `user_cancelhold` command cancels a hold operation previously performed on a user’s certificates (one or both of the user’s certificates were revoked with the reason On Hold). The user’s certificates become valid again once the `user_cancelhold` operation takes effect.

## Syntax

```
user_cancelhold <userID> <serialNumber> E|V Y|N
```

Where:

- <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).
- <serialNumber> is the serial number of the certificate.
- `E|V` cancels the hold on either the encryption (`E`) or verification (`V`) certificate.
- `Y|N` specifies whether to issue the CRL immediately. Enter `Y` to issue the CRL immediately or enter `N` to issue the CRL later.

## user\_cancelrecover

The `user_cancelrecover` command returns a user to the Active state if the user is set for key recovery. The user must be in the Key Recovery state for this command to work.

### Syntax

```
user_cancelrecover <userID>
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).

## user\_convertv1

The `user_convertv1` commands converts a V2 user (a user with a V2 profile) to a V1 user (a user with a V1 profile).

### Syntax

```
user_convertv1 <userID>
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).

Before completing the conversion, Security Manager checks that the certificate type assigned to the user is compatible with the V1 client type. To be compatible, the certificate type must have a verification certificate definition. If the user is not a 1-key-pair user, the certificate type should also have an encryption certificate definition. If it does not, there will be an error the next time the V1 client application attempts to get a new encryption certificate.

In most cases, a key recover is required to complete the conversion. Use the `user_recover` command to recover the user (see [“user\\_recover” on page 667](#)).

## user\_createdirenty

The `user_createdirenty` command creates an entry in the directory. There are two methods for creating directory entries using this command:

- Template method

With the Template method, the DN of the entry is not specified when the user object is declared. Instead, the DN of the entry is constructed from the attributes that are identified in the template as mandatory for the DN, and from any other attributes that are specified using the `user_addtodn` command. The template used to create the entry is defined using the `user_settemplate` command. The attributes that are assigned values using the `user_setattribute` command are also added to the directory entry.

If the state of the user object identified by the user ID is such that it does not meet the requirements of the template (for example, not all required attributes are set), then the directory entry is not created and an error is returned.

- Custom method

With the Custom method, the DN of the entry is specified when the user object is declared. Therefore, it is not required to specify a template. Attributes, including the `objectClass` attribute, are assigned values using the `user_setattribute` command. If the attributes specified are not consistent with both the DN and the directory schema, then the DN is not created and an error is returned.

### Syntax

```
user_createdirectory <userID>
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).

## user\_deactivate

The `user_deactivate` command deactivates the user identified by user ID, putting the user in the Deactivated state. To be deactivated, the user must be in the Active, Key Recovery, or Added state.

To reactivate a user in the Deactivated state, use the `user_reactivate` command (see [“user\\_reactivate” on page 667](#)).

### Syntax

```
user_deactivate <userID>
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).

## user\_deletedirectory

The `user_deletedirectory` command deletes the directory entry for the user identified by the user ID.

## Syntax

```
user_deletedirentary <userID>
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).

## user\_export

The `user_export` command allows you to start, finish, or cancel a user export operation. You cannot perform more than one user export operation (start, finish, or cancel) with the same `user_export` command. For example, you can start a user export operation with the `user_export` command, but you cannot start and finish the export operation with the same command.

For more information about exporting and importing users, see [“Moving users to a new Certification Authority” on page 202](#).

Ensure there is a trust relationship established between the old and new Certification Authority (CA) before you move a user from the old CA to the new CA.

---

**Note:** You cannot export roaming users. Attempting to exporting roaming users results in errors. To export roaming users, you must first change them into desktop users.

---

This section includes syntax and descriptions for:

- [“Starting a user export operation” on page 664](#)
- [“Finishing the user export operation” on page 665](#)
- [“Canceling a user export operation” on page 665](#)

## Starting a user export operation

A user export operation is started by exporting the user's key history to an export file.

When processing the `user_export` command, the bulk commands necessary to import the user into another CA are written to a file. The name of the file is the same as the file that issued the `user_export` command, with `import` appended before the extension. For example, if the file containing the `user_export` command is called `moveusers.entra`, then the new file is called `moveusers_import.entra`.

The following information is written to the file:

- The user's new DN.  
This is specified as part of a user command. If the user's new DN is not provided, then the user's original DN is used.
- The user's exported key history blob.  
This is included using the `user_import` command:



```
user_import <userID> { key_history_blob }
```

The exported key history blob is an encoded and encrypted string of information (much like the private keys in an EPF file). To keep line lengths to a minimum, the key history blob is broken up into multiple lines of 80 characters each. The entire blob argument is enclosed inside curly braces to ensure that the Tcl interpreter does not do any argument substitution in the encoded string.

The commands written to the file are necessary but not sufficient to import the user into another CA. An administrator at the importing CA must edit this file to specify CA-specific information.

After a user's key history is exported, the user is placed in the Export Hold state at the exporting CA.

### Syntax

```
user_export <userID> start <importCADN> [<newDN>]
```

Where:

- `<userID>` is the identifier of the user object. The user ID was specified in the `user` command (see ["user" on page 659](#)).
- `<importCADN>` identifies the distinguished name (DN) of the importing CA.
- `<newDN>` is the new DN of the user when imported into the new CA (often the DN is changed to reflect that the user has moved to and belongs to a new CA). If this argument is not provided, it is assumed that the DN is unchanged.

## Finishing the user export operation

Once the user is successfully imported into the new CA, the old CA needs to complete the export operation by moving the user from the Export Hold state to the Export state. This is done using the `user_export` command with the `finish` argument.

### Syntax

```
user_export <userID> finish
```

Where `<userID>` is the identifier of the user object. The user ID was specified in the `user` command (see ["user" on page 659](#)).

## Canceling a user export operation

If you need to cancel the `export` operation (for example, the wrong user was exported by mistake), you can move the user from the Export Hold or Export state to whatever state the user was in when the export operation was started. To do this, use the `user_export` command with the `cancel` subcommand argument.

## Syntax

```
user_export <userID> cancel
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).

## user\_free

The `user_free` command frees the resources associated with a user object. Use this command in a bulk script once the user object identified by the user ID is no longer required. This saves memory if many unique user IDs are used in a bulk script.

## Syntax

```
user_free <userID>
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).

## user\_import

The `user_import` command adds to Security Manager the user identified by the user ID, and imports the provided key history.

The `user_import` command is written to an export file when a user is exported. You should never manually code or edit this command in a bulk script.

The key history blob is an encoded and encrypted string of information (much like the private keys is an EPF file). The entire blob is enclosed in curly braces so that the Tcl interpreter does not perform any variable substitution in the encoded string.

For information about importing users, see [“Moving users to a new Certification Authority” on page 202](#).

## Syntax

```
user_import <userID> { key_history_blob }
```

Where:

- <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).
- <key\_history\_blob> is the encoded key history for the user.

## user\_notifyclient

The `user_notifyclient` command notifies the client application that a change to a user's certificate definitions occurred for the user identified by the user ID. For more information about notifying clients, see [“Notifying client applications” on page 242](#).

## Syntax

```
user_notifyclient <userID>
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see ["user" on page 659](#)).

## user\_reactivate

The `user_reactivate` command reactivates the user identified by the user ID. The user must be in the Deactivated state to be reactivated.

## Syntax

```
user_reactivate <userID>
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see ["user" on page 659](#)).

## user\_recover

The `user_recover` command sets up the user identified by the user ID for key recovery. This command requires that the user be in the Active state. If the Entrust PKI administrator processing the bulk script has permission to view activation codes, then the activation codes appear in the bulk output log file.

## Syntax

```
user_recover <userID>
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see ["user" on page 659](#)).

## user\_reissueactivationcodes

The `user_reissueactivationcodes` command reissues the activation codes for the user identified by user ID. The user must be in the Added or Key Recovery state. If the Entrust PKI administrator processing the bulk script has permission to view activation codes, then the activation codes appear in the bulk output log file.

Reissuing activation codes generates new codes with new creation and expiry dates. The validity of the previous activation codes does not affect this procedure. You can reissue new activation codes before the old activation codes expire (for example, when a user's activation codes are lost or stolen). For more information about activation code lifetimes, see ["Configuring the lifetime of activation codes" on page 154](#).

## Syntax

```
user_reissueactivationcodes <userID>
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see ["user" on page 659](#)).

## user\_renamedirentry

The `user_renamedirentry` command renames the directory entry identified by the user ID. If the directory entry is also an Entrust PKI user, then you should update Security Manager using the `user_assigndn` command (see ["user\\_assigndn" on page 661](#)). Your directory must be LDAPv3-compliant.

## Syntax

```
user_renamedirentry <userID> <newDN>
```

Where:

- <userID> is the identifier of the user object. The user ID was specified in the `user` command (see ["user" on page 659](#)).
- <newDN> is the user's new DN.

## user\_restoretodir

The `user_restoretodir` command restores to the directory the Entrust-specific information for the user identified by the user ID. This command only restores attribute information, so the directory entry for the user ID must already exist.

You should back up your directory on a regular basis using your directory backup tools. The `user_restoretodir` command only restores certificate information. If any other type of directory information becomes corrupt or is lost (such as object classes, directory entries, or directory attributes), you can only retrieve this information from a backup generated by your directory backup tools. For more information about backing up your directory, see the *Security Manager Operations Guide*.

## Syntax

```
user_restoretodir <userID>
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see ["user" on page 659](#)).

## user\_retrieve

The `user_retrieve` command retrieves the key history for the user identified by the user ID from the Security Manager archive. For the `user_retrieve` command to work, the user must already exist in the directory as a non-Entrust PKI user. This

command also commits any user properties that are set with the `user_setproperty` command.

### Syntax

```
user_retrieve <userID>
```

Where `<userID>` is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).

## user\_revoke

The `user_revoke` command revokes certificates for the user identified by the user ID. Which certificates are revoked, and for what reason, are controlled using the command parameters.

### Syntax

```
user_revoke <userID> [<comment>] [<reason>] [All|Latest]  
[E|V|Both] [<last_uncompromised_date>]
```

Where:

- `<userID>` is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).
- `<comment>` is any text you want to associate with the revoked certificates. For example, you can describe why you are revoking the user's certificates. If you do not specify a comment, the comment defaults to “No comment”.
- `<reason>` provides the reason for revoking the certificates. Valid reasons are: `U` (Unspecified), `KC` (Key Compromise), `AC` (Affiliation Change), `S` (Superseded), `COO` (Cessation Of Operation), `OH` (On Hold). If you do not specify a reason, it defaults to Unspecified.
- `All|Latest` are options that specify whether to revoke all certificates (`All`) or the most recent certificates (`Latest`). If not specified, all certificates are revoked.
- `E|V|Both` are options that specify whether to revoke only the encryption certificates (`E`), only the verification certificates (`V`), or both the encryption and verification certificates (`Both`). If not specified, both certificates are revoked.
- `<last_uncompromised_date>` is the date the certificates were last known to be uncompromised. This argument is only required if `<reason>` is `KC` (Key Compromised).

The format of the date is MM/DD/YYYY (for example, 12/25/2007). If you do not specify a date and `<reason>` is `KC`, or if the date precedes the issue

date of the revoked certificate, then the date defaults to the issue date of the revoked certificate.

## user\_setattribute

The `user_setattribute` command adds or deletes values from a user's directory attribute. If the user ID represents an existing user in the directory, then the specified attribute modification is made immediately. If the user ID represents a new user that does not yet exist in the directory, then the specified attribute modification is cached until you enter the `user_createdirententry` command.

### Syntax

```
user_setattribute <userID> <attribute> +|- <value>
```

Where:

- `<userID>` is the identifier of the user object. The user ID was specified in the `user` command (see ["user" on page 659](#)).
- `<attribute>` is the name of the attribute.
- `+|-` specifies whether to add a value to the attribute (+) or delete a value from an attribute (-).

To delete all values from a directory attribute, enter an asterisk (\*) for `<value>`.

- `<value>` is the value of the directory attribute.

## user\_setparentdn

The `user_setparentdn` command specifies the parent distinguished name (DN) to create a new directory entry under when the `user_createdirententry` command is issued for the user ID. This command is effective only when creating a directory entry using a template. The parent DN must already exist in the directory. In the absence of this command, the user's directory entry is created under the DN of the CA.

### Syntax

```
user_setparentdn <userID> <parentDN>
```

Where:

- `<userID>` is the identifier of the user object. The user ID was specified in the `user` command (see ["user" on page 659](#)).
- `<parentDN>` is the DN under which the user is created.

## user\_setproperty

The `user_setproperty` command modifies different user properties, such as the user's role or group membership. You can only modify a single user property at time. For example, you can only modify the user's role or group membership in one command. To modify more than one user property, you must enter the `user_setproperty` command more than once.

Any property modifications made with the `user_setproperty` command are not applied until the `user_applyproperties` command is processed (see ["user\\_applyproperties" on page 660](#)).

This section includes syntax and descriptions for

- ["Setting certificate category and type" on page 671](#)
- ["Setting group membership" on page 672](#)
- ["Setting the subjectAltName \(alternate identity\)" on page 672](#)
- ["Setting key update options—key lifetimes" on page 673](#)
- ["Setting key update options—key expiry" on page 673](#)
- ["Setting certificate extension variable values" on page 674](#)
- ["Setting user role" on page 675](#)
- ["Setting encryption OIDs" on page 675](#)
- ["Setting verification OIDs" on page 676](#)

### Setting certificate category and type

The `user_setproperty` command sets properties for the user identified by the user ID. The `user_setproperty` command can set the certificate category and type for the user identified by the user ID. If you do not specify a certificate type when adding or importing a user into Security Manager, Security Manager Administration assigns the user the default certificate category (Enterprise) and certificate type (ent\_default). There are no defaults for users who are already activated as Entrust PKI users. For a list of the built-in certificate types, see ["Predefined certificate types" on page 535](#).

#### Syntax

```
user_setproperty <userID> certificate_type <category> <type>
```

Where:

- `<userID>` is the identifier of the user object. The user ID was specified in the `user` command (see ["user" on page 659](#)).
- `<category>` identifies the category of the certificate type.
- `<type>` identifies the certificate type.

## Setting group membership

The `user_setproperty` command sets properties for the user identified by the user ID. The `user_setproperty` command can add and remove groups from the user identified by the user ID. A user can belong to more than one group, but must belong to at least one group. A group must already exist before you can add a user to that group.

No warning or error occurs if you add the user to a group and the user already belongs to that group. No warning or error occurs if you remove the user from a group and the user does not belong to that group.

### Syntax

```
user_setproperty <userID> group +|- <groupname>
```

Where:

- `<userID>` is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).
- `+|-` specifies whether to add the user to a group (+) or to remove the user from a group (-).
- `<groupname>` is the name of the group to which to add or remove the user. You can enter an asterisk (\*) for the group name to add the user to all current and future groups, or to remove the user from all groups. Users must belong to at least one group. If you remove the user from all groups, you must immediately add the user to a group with another `user_setproperty` command.

## Setting the subjectAltName (alternate identity)

The `user_setproperty` command sets properties for the user identified by the user ID. The `user_setproperty` command can specify the value of the `subjectAltName` (also called an alternate ID) property for the user identified by the user ID.

### Syntax

```
user_setproperty <userID> alternate_id +|- [<value>]
```

Where:

- `<userID>` is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).
- `+|-` specifies whether to add a value to an attribute (+) or delete a value from an attribute (-).
- `<value>` is the value to append to or remove from the current alternate ID.



If no value is provided and the previous argument is the subtraction operator (-), then the alternate identity is cleared. To create a value that you can enter here, see [Step 13 on page 291](#).

## Setting key update options—key lifetimes

The `user_setproperty` command sets properties for the user identified by the user ID. The `user_setproperty` command can set the key lifetimes for the user identified by the user ID.

If this command-property combination is given without any of the lifetime arguments, then the user's key update options are set to use the CA's default key update options.

Do not use the `key_lifetimes` property argument if you already used the `key_expiry` property argument. Users can only have their keys set with a lifetime or an expiry date. The `key_lifetimes` property argument will overwrite a previously entered `key_expiry` property argument.

For more information about key update options, see [“Configuring user key update options” on page 230](#).

### Syntax

```
user_setproperty <userID> key_lifetimes [<PubEncKey_life>  
<PubVerKey_life> <PrvSignKey_life>]
```

Where:

- `<userID>` is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).
- `<PubEncKey_life>` is the encryption public key lifetime in months (from 2–420).
- `<PubVerKey_life>` is the verification public key lifetime in months (from 2–420).
- `<PrvSignKey_life>` is the signing private key lifetime expressed as a percentage of the public verification key lifetime (from 1–100).

## Setting key update options—key expiry

The `user_setproperty` command sets properties for the user identified by the user ID. The `user_setproperty` command can set the key expiry dates for the user identified by the user ID.

If this command-property combination is given without any of the lifetime arguments, then the user's key update options are set to use the CA's default key update options.

Do not use the `key_expiry` property argument if you already used the `key_lifetimes` property argument. Users can only have their keys set with a lifetime or an expiry date. The `key_expiry` property argument will overwrite a previously entered `key_lifetimes` property argument.

For more information about key update options, see [“Configuring user key update options” on page 230](#).

For active users, you must set the users up for key recovery before you set their key expiry dates. See [“user\\_recover” on page 667](#).

## Syntax

```
user_setproperty <userID> key_expiry [<PubEncKey_PrivSignKey_expiry  
PubVerKey_expiry>]
```

Where:

- `<userID>` is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).
- `<PubEncKey_PrivSignKey_expiry>` is the expiry date and time for the encryption public and signing private keys.

The format for the date is dependent on regional settings. For example, the default for English (US) is MM/DD/YYYY. Enter the time as hh:mm:ss followed by AM or PM.

The date and time must occur at least 12 hours into the future. The date and time cannot exceed the expiry date of the CA's signing certificate. The date and time contains spaces so you must enclose the date and time in quotation marks.

- `<PubVerKey_expiry>` is the expiry date and time for the verification public key.

The format for the date is dependent on regional settings. For example, the default for English (US) is MM/DD/YYYY. Enter the time as hh:mm:ss followed by AM or PM.

The date and time must occur at least 12 hours into the future. The date and time cannot exceed the expiry date of the CA's signing certificate. The date and time must be later than `<PubEncKey_PrivSignKey_expiry>`. The date and time contains spaces so you must enclose the date and time in quotation marks.

## Setting certificate extension variable values

The `user_setproperty` command sets properties for the user identified by the user ID. The `user_setproperty` command can assign values to the certificate extension variables that are available for the user's certificate type. You can assign only one

value to a certificate extension variable. If the specified variable already has a value, then the value is replaced.

Before you specify certificate extension variables, you must specify the certificate type (see [“Setting certificate category and type” on page 671](#)). You must specify the certificate type even if you do not plan to change the certificate type.

## Syntax

```
user_setproperty <userID> certificate_extension <variable> <value>
```

Where:

- `<userID>` is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).
- `<variable>` is the name of the variable to assign a value to.
- `<value>` is the value to assign to the extension.

## Setting user role

The `user_setproperty` command sets properties for the user identified by the user ID. The `user_setproperty` command can set the role assigned to the user. You can assign only one role to a user at a time. Assigning a user a new role replaces the user's old role.

## Syntax

```
user_setproperty <userID> role <roleID>
```

Where:

- `<userID>` is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).
- `<roleID>` is the name of the role to assign to the user.

You can assign the user any of the predefined roles (see [“Predefined Security Manager user roles” on page 354](#)) or any custom role created for your organization. You must surround any `roleID` made up of two words or more with quotation marks ("" ) in order for the Tcl interpreter to read the multiple words as a single argument.

## Setting encryption OIDs

The `user_setproperty` command sets properties for the user identified by the user ID. The `user_setproperty` command can assign a custom set of encryption object identifiers (OIDs) to the user identified by the user ID.

For more information about assigning a custom set of encryption OIDs to a user, see [“Configuring user encryption and verification OIDs” on page 237](#).

## Syntax

```
user_setproperty <userID> enc_oids [+|- [<value>]]
```

Where:

- <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).
- +|- specifies whether to add encryption OIDs to a user (+) or remove encryption OIDs from a user (-).
- <value> is one or more encryption OIDs.

The available OIDs you can choose from must already exist. See [“Adding OIDs” on page 90](#) for information about creating OIDs. You can only assign a maximum of 10 encryption OIDs to users. To specify two or more encryption OIDs, separate each OID with a space. For example: 1.1.1.1 1.1.1.2 1.1.1.3.

If you are adding encryption OIDs to a user (+ <value>) and you specify an encryption OID that is already assigned to the user, the OID is ignored. If you are removing encryption OIDs from a user (- <value>) and you specify an encryption OID that is not assigned to the user, the OID is ignored.

If no value is provided and no operators (+ or -) are provided, then only the default encryption OIDs are assigned to the user.

If no value is provided and the previous argument is the subtraction operator (-), then all encryption OIDs are removed, including all default encryption OIDs.

## Setting verification OIDs

The `user_setproperty` command sets properties for the user identified by the user ID. The `user_setproperty` command can assign a custom set of verification object identifiers (OIDs) to the user identified by the user ID.

For more information about assigning a custom set of verification OIDs to a user, see [“Configuring user encryption and verification OIDs” on page 237](#).

## Syntax

```
user_setproperty <userID> ver_oids [+|- [<value>]]
```

Where:

- <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).
- +|- specifies whether to add verification OIDs to a user (+) or remove verification OIDs from a user (-).
- <value> is one or more verification OIDs.

The available OIDs you can choose from must already exist. See [“Adding OIDs” on page 90](#) for information about creating OIDs. You can only assign a maximum of 10 verification OIDs to users. To specify two or more verification OIDs, separate each OID with a space. For example: 1.1.1.1 1.1.1.2 1.1.1.3.

If you are adding verification OIDs to a user (+ <value>) and you specify a verification OID that is already assigned to the user, the OID is ignored. If you are removing verification OIDs from a user (- <value>) and you specify a verification OID that is not assigned to the user, the OID is ignored.

If no value is provided and no operators (+ or -) are provided, then only the default verification OIDs are assigned to the user.

If no value is provided and the previous argument is the subtraction operator (-), then all verification OIDs are removed, including all default verification OIDs.

## user\_settemplate

The `user_settemplate` command sets the template for the user identified by the user ID. This command is only required when creating a new directory entry using a template.

### Syntax

```
user_settemplate <userID> <template>
```

Where:

- <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).
- <template> is the name of the user type template.

## user\_updatekeypairs

The `user_updatekeypairs` command updates the key pairs for the user identified by the user ID. When a user's key pairs are updated, the user's profile is updated with the new keys the next time the user logs in. You cannot cancel an update key pairs operation.

For more information about updating key pairs, see [“Updating key pairs” on page 240](#).

---

**Note:** If you have set the user encryption key to RSA-4096 or RSA-6144, the key update operation is considerably slower than for RSA-1024 or RSA-2048. See the *Security Manager Operations Guide* for more information.

---

## Syntax

```
user_updatekeypairs <userID>
```

Where <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).

## user\_refresh\_subaltname\_from\_dir

The `user_refresh_subaltname_from_dir` command refreshes the `subjectAltName` component values of existing users based on changes in the directory.

### Syntax:

```
user_refresh_subaltname_from_dir <userID> <refreshMode>
```

Where:

- <userID> is the identifier of the user object. The user ID was specified in the `user` command (see [“user” on page 659](#)).
- <refreshMode> is one of:
  - `replace`: deletes the existing `subjectAltName` components and creates new ones based on directory values.
  - `update` (default): allows you to keep the `subjectAltName` component values that you have added through other means than auto-population. Any component that is mapped to a directory attribute is replaced with new information from the directory.
  - `append`: adds new components and does not modify existing components or their values.

## Entering international characters in distinguished names

If all the distinguished names (DNs) in your Entrust infrastructure are expressed adequately in ASCII, you do not need to read this appendix.

If you are trying to enter or display DN's using an ASCII keyboard or monitor, but these DN's contain characters that are outside the ASCII character set, you need to read this appendix. For example, if you require Latin-1 characters, Chinese, Japanese, and so on, and these characters that you may need to display on ASCII monitors or input from ASCII keyboards, this appendix will be of interest.

This appendix includes the following sections:

- [“What is internationalization support?” on page 680](#)
- [“When to use international characters” on page 681](#)
- [“How to enter international characters in Security Manager Administration” on page 683](#)

---

**Note:** If you are planning to use international characters in a CA DN, contact Entrust Support for assistance.

---

# What is internationalization support?

Support for applications that can run anywhere in the world requires support for as wide a variety of languages as possible. ISO 10646 and Unicode have addressed the problem of supporting different languages by creating an international character set, Universal Multiple-Octet Coded Character Set (UCS).

For more information about Unicode, see <http://www.unicode.org>.

Security Manager supports Unicode by using ASN.1 UTF8String and BMPString. UTF8 is also used with LDAP version 3 and most browsers. This format allows representation of the 16-bit Unicode characters by a series of 8-bit characters.

One benefit of UTF8 is that ASCII characters are not changed when represented as UTF8. UTF8 is defined by RFC 2279. For more information about UTF8, see <http://www.cis.ohio-state.edu/htbin/rfc/rfc2279.html>.

International characters are displayed in Security Manager using an ASCII escape mechanism that is described in RFC 2253. Through this escape mechanism, up to nine ASCII characters are used to represent a single Unicode character. Unicode characters are used in Security Manager to implement international character support.

International names for groups, roles, searchbases, and policy certificates can contain at least 25 Unicode characters.



# When to use international characters

You may use international characters when entering information that affects a user's distinguished name (DN) or their directory attributes. You may also use international characters when entering other information in Security Manager Administration.

Topics in this section:

- [“Attributes that support international characters” on page 681](#)
- [“Using international characters during installation” on page 681](#)
- [“Directory issues with international characters” on page 682](#)

## Attributes that support international characters

In Security Manager, you can use international characters in attribute values that allow Unicode characters.

Any attribute that uses the directoryString syntax (X.520 standard) can have a Unicode value.

For example, you can use international characters while you are logged in to Security Manager Administration as an Entrust PKI administrator to perform the following tasks:

- adding an Entrust PKI user
- finding an Entrust PKI user
- adding an attribute to a DN
- changing an attribute in a DN

For more information about these tasks, see [“Administering users” on page 129](#).

Bulk operations also support the use of international characters. For information about bulk operations, see [“Performing bulk operations” on page 275](#).

## Using international characters during installation

You can also use international characters during installation of Security Manager when you are prompted for the following directory information:

- CA DN
- First Officer DN
- Directory Administrator DN
- Superior CA DN

---

**Note:** If you are planning to use international characters in a CA DN, contact Entrust Support for assistance.

---

For more information about installing Security Manager, see the *Security Manager Installation Guide*.

Security Manager Administration writes data to your directory. You may have other applications writing to the directory. When you enter international characters from Security Manager Administration, they are written to the directory and stored there in UTF8 format.

International characters are used most often with the common name (cn=) attribute. It is also possible to create a new CA with international characters used in the organization name (o=) and organizational unit name (ou=) attributes.

## Directory issues with international characters

In order to use international characters you must use a directory that supports LDAP version 3. In a cross-certified environment, all the directories must support LDAP version 3.

The directory attribute `rfc822Mailbox` does not support international character sets. The `rfc822Mailbox` attribute supports Internet mail addresses, which only allow ASCII characters.

# How to enter international characters in Security Manager Administration

There are two ways you can enter international characters in Security Manager Administration:

- by typing escaped character strings on an ASCII keyboard
- by typing international characters when using the Microsoft Input Method Editor (IME)

---

**Note:** Internet email addresses must be ASCII text. As a result, you must enter all email addresses in the English form of the name, for example, pat.jones@entrust.com.

---

Topics in this section:

- [“Entering international characters from an ASCII keyboard” on page 683](#)
- [“Entering international characters using the IME” on page 684](#)
- [“How Security Manager Administration displays international characters” on page 685](#)

## Entering international characters from an ASCII keyboard

There are two ways to enter international characters from an ASCII keyboard:

- inputting the characters in a dialog box (such as an installation dialog box or a dialog box in Security Manager Administration) ([“To enter international characters in a dialog box” on page 683](#))
- inputting the characters in a bulk script ([“To enter international characters in a bulk script” on page 684](#))

### To enter international characters in a dialog box

- 1 Obtain RFC 2253 and RFC 2279 from the Internet.
- 2 Determine the UTF8 values of the DN attribute you wish to enter.  
UTF8 values are generated from Unicode values. One method to generate them is described in RFC 2279 in section 2, “UTF8 definition.”
- 3 Enter the escaped RFC 2253 representation of the UTF8 data in Security Manager Administration.

For example, when creating a new user, the **Last Name** field in the New User dialog box creates an “sn =” entry in the directory. If you wanted to enter a user with the last name of “Entrust” (or “Entorasuto”) in the New User dialog box

and have this name displayed in Japanese on a Japanese monitor, you would enter the following in the **Last Name** field:

`\e3\82\xa8\e3\83\b3\e3\83\88\e3\83\xa9\e3\82\b9\e3\83\88`

---

**Note:** The backslash (\) is used as an escape character. You type in actual backslashes when you are entering the escaped RFC 2253 representation of the international character or symbol using an ASCII keyboard. If you want a backslash to appear in the data, type a double backslash: \\. This is particularly important during bulk operations, where the backslash is used to precede the double quote symbol (\") and optionally the plus sign (\+) as an escape character.

---

### To enter international characters in a bulk script

- 1 If you have not already done so, obtain RFC 2253 and RFC 2279 from the Internet and determine the UTF8 values of the DN attributes you wish to enter as described in [“To enter international characters in a dialog box” on page 683](#).
- 2 In a bulk script, enter the escaped RFC 2253 representation of the UTF8 data when declaring parts of the user’s DN.

For example, suppose you want to enter a new user named “Tim Entorasuto” in a bulk script. If you want the DN of this user to display on a Japanese monitor, you enter the bulk input as follows, using double backslashes “\\” inside and outside the quotation marks:

```
user foo
user_settemplate foo Person
user_setattribute foo cn + "Tim\
\\e3\\82\\a8\\e3\\83\\b3\\e3\\83\\88\\e3\\83\\a9\\e3\\82\\b9\\e3\\
83\\88"
user_setattribute foo sn + \
"\\e3\\82\\a8\\e3\\83\\b3\\e3\\83\\88\\e3\\83\\a9\\e3\\82\\b9\\e3\\
\\83\\88"
user_createdirentree foo
```

## Entering international characters using the IME

When using the Microsoft Input Method Editor (IME) to enter information, you can type the characters or ideographs of a language directly at the keyboard. For example, if you are running Security Manager Administration on a Japanese version of Windows, you can use the Japanese Input Method Editor (IME) to enter Japanese characters directly into the attribute fields.

Security Manager Administration automatically converts between the local language and escaped RFC 2253 if possible. The same conversions occur for users who access applications that are created with Security Manager Administration Toolkit.

Local characters or ideographs are translated from the Microsoft Multi-Byte Character Set (MBCS) format to UTF8 format (the globalized character set) as they are sent to the directory. When sent from Security Manager Administration to Security Manager, these characters are translated from UTF8 to RFC 2253 escape sequences.

## How Security Manager Administration displays international characters

The characters that you enter into Security Manager Administration on a Japanese system are automatically converted by Security Manager Administration to UTF8 format before they are stored in the directory. When you want to display the Japanese DN and its attributes in Security Manager Administration, it is automatically converted back from UTF8 format to the local character set.

Note that it is not always possible to perform these conversions. For example, take the case of a user in Tokyo who is added to Entrust by an Entrust PKI administrator in Tokyo using a Japanese version of Windows. In this case, the user's name is entered and displayed in Japanese. If an Entrust PKI administrator in New York views this user on an English system, the user's name cannot display in Japanese. In this case, it displays in the escaped RFC 2253 format.

In either case, you can still administer the user correctly, but the Entrust PKI administrator on the English machine sees the escaped RFC 2253 format of the name. The characters are stored correctly for Japanese display. Entrust PKI administrators at an English monitor do not see an English translation of the Japanese name.



## Glossary of terms

Term	Definition
<b>activation codes</b>	When an Entrust PKI administrator (such as a Security Officer) adds a user in Security Manager Administration, a reference number and authorization code are generated. Together, the reference number and authorization code are called activation codes. A users enters this information in Entrust client software to create a profiles. See <a href="#">profile</a> .
<b>Administration Service Handler (ASH) subsystem</b>	A part of the Security Manager service that handles administrative requests from Security Manager Administration and other applications that use the Security Manager Administration toolkit.
<b>ASH subsystem</b>	See <a href="#">Administration Service Handler (ASH) subsystem</a> .
<b>ARL</b>	See <a href="#">authority revocation list (ARL)</a> .
<b>attribute</b>	A directory term. Each piece of information that describes an aspect of a directory entry is called an attribute. Entries are the building blocks of the directory. An attribute comprises an attribute type and at least one attribute value. An example of an attribute type is telephone number, with the number 555 1212 considered an attribute value.
<b>audit logs</b>	All significant transactions in Security Manager, such as user initializations and changes in users' information, are recorded in audit logs. You can use audit logs for troubleshooting.
<b>authentication</b>	The process of proving your identity. In a Security Manager system, authentication works through a password-protected encrypted file, called the Entrust profile. The Entrust profile—among other important data—contains a user's identity (DN), decryption private keys, their signing keys, and the CA signing keys. When users log into a Security Manager client application, they choose their Entrust profile and enter their password. This process verifies their identity and allows them to access their private data.

Term	Definition
<b>authority revocation list (ARL)</b>	A signed, time-stamped list of the serial numbers of CA public key certificates (including cross-certificates) that are revoked. It is signed with the signing key of the CA that issued the certificates.
<b>authorization code</b>	An alphanumeric code (for example, CMTJ-8VOR-VFNS), obtained from an Entrust PKI administrator. It is required along with its corresponding reference number to create a new Entrust profile or to recover an existing profile. An authorization code and its corresponding reference number are called activation codes. Authorization codes can only be used once.
<b>backup</b>	A backup contains information about the Security Manager database and its content at a specific point in time (the time that the backup was performed). A backup enables you to restore the Security Manager database and its content if a problem occurs.
<b>bulk processing</b>	In Security Manager, batch processing is called bulk processing. It is the process of running operations on multiple users at one time.
<b>CA</b>	See <a href="#">Certification Authority (CA)</a> .
<b>CA directory password</b>	Security Manager uses the CA directory password to access and modify the directory. It is recommended that you change this value periodically. You must change the password if you suspect that it is compromised.
<b>CA domain</b>	The people (in a company, work group, educational or governmental institution), the processes, and tools that operate under the same software license and belong to the same Certification Authority (CA). See <a href="#">Certification Authority (CA)</a> .
<b>CA profile</b>	<p>The CA has its own profile, just like a user does. This profile, called <code>ca.epf</code>, is created during the initialization of Security Manager. Two key pairs are associated with the CA profile: a signing key pair, called the CA user signing key pair and an encryption key pair, called the CA user encryption key pair.</p> <p>Security Manager also creates a random password for the CA profile during initialization and stores the password securely in the Security Manager database. You must recover the CA profile if it is damaged. There should be no reason to suspect that the CA profile password has been compromised since it is known only to Security Manager and is stored securely in the Security Manager database.</p>
<b>CA signing key pair</b>	The signing key pair of the CA, which comprises the CA signing private key and CA verification public key. The CA signing private key digitally signs client certificates, and the CA verification public key verifies signatures. See also <a href="#">signing key pair</a> .
<b>CA signing private key</b>	The private key portion of the CA signing key pair. The CA signing private key is used to digitally sign client certificates which can be verified with the CA verification public key. A CA signs all certificates it issues using the CA signing key. See also <a href="#">signing key pair</a> .



Term	Definition
<b>CA verification public key</b>	The public key portion of the CA signing key pair. It verifies client certificates that are signed by the CA signing private key.
<b>certificate</b>	<p>A collection of publicly available information in standard format about an entity that is digitally signed by the <a href="#">Certification Authority (CA)</a>. A certificate is used to uniquely identify people and resources over networks such as the Internet. Certificates also enable secure, confidential communication between two parties.</p> <p>A certificate typically includes a variety of information pertaining to its owner and to the CA that issued it, including the name of the holder and other holder identification information (for example, the URL of the Web server using the certificate or an email address), the holder's public key, the name of the Certification Authority that issued the certificate, a serial number, and the validity period (or lifetime) of the certificate (a start and an end date).</p> <p>The CA issues all certificates according to the format and structure of the X.509 version 3 standard.</p>
<b>certificate expiration date</b>	The date after which a user's <a href="#">certificate</a> should no longer be trusted.
<b>certificate revocation</b>	<p>A user's encryption public key certificate and verification public key certificate must be revoked if a user's encryption public key or signing key pair is no longer trusted for any reason. For example, you may need to revoke a certificate if you suspect that the user's decryption private key and signing private key disclosed, or if the user's Entrust password has been compromised.</p> <p>An organization's security policy may also require that certificates be revoked when users leave the organization or when their distinguished names change. When certificates are revoked, Security Manager updates the appropriate certificate revocation list with information about the revoked certificates.</p>
<b>certificate revocation list (CRL)</b>	A signed and timestamped <a href="#">certificate</a> containing the serial numbers of public key certificates that have been revoked, and a reason for each revocation.
<b>Certification Authority (CA)</b>	<p>The part of <a href="#">Security Manager</a> that ensures the trustworthiness of electronic identities. It issues electronic identities in the form of public key certificates, policy certificates, cross-certificates, certificate revocation lists (see <a href="#">certificate revocation list (CRL)</a>), and authority revocation lists, and signs the certificates with its signing key to ensure the integrity of the electronic identity.</p> <p>See <a href="#">certificate</a>.</p>

Term	Definition
<b>changing DNs</b>	<p>Once a user is activated you can change their DN for any number of reasons (for example, when the user's name is legally changed or if the user moves to a different department or location).</p> <p>You cannot change the DN of user who is in the added state (that is, not activated yet) or who is set up for key recovery.</p>
<b>Chip Authentication</b>	<p>This is one of the steps taken in <a href="#">Extended Access Control (EAC)</a>. It is the process of authenticating the ePassport chip to the <a href="#">Inspection System</a> terminal. A successful authentication proves that the chip is genuine.</p> <p>This is followed by <a href="#">Terminal Authentication</a>.</p>
<b>client application</b>	An application running as a desktop agent that receives information from a server application and requests a service provided by the server application.
<b>combined CRL</b>	A single, large Certificate Revocation List (CRL). It contains all revoked certificates that were issued by the CA. These are useful for compatibility with third-party applications.
<b>Country Verifying Certification Authority (CVCA)</b>	<p>A country's CA. A country can have only one CVCA. It interacts with Document Verifiers (DVs) of the same country and of foreign countries. It certifies the biometric information stored on the ePassports that the country issues to its citizens. It determines which DVs (domestic or foreign) are allowed access to that information.</p> <p>See also <a href="#">Document Verifier (DV)</a> and <a href="#">e-passport</a>.</p>
<b>CRL</b>	See <a href="#">certificate revocation list (CRL)</a> .
<b>CRL distribution points</b>	Security Manager maintains multiple CRLs. It does so at unique distribution points in the directory because it is not possible to have only one CRL for all certificates in a CA domain, in most CAs. In Security Manager, one CRL can list up to 750 certificates. Each user's certificate contains a pointer to its appropriate distribution point in the directory so that Entrust client software knows where to find the corresponding CRL for a given certificate.
<b>cross-certificates</b>	When two CAs cross-certify, each signs the other's verification public key certificate using its own signing private key. The result is two cross-certificates.
<b>cross-certification</b>	A process by which two CAs securely exchange keying information so that each can effectively certify the trustworthiness of the other's keys. Once two (or more) domains have cross-certified, users within those domains can validate each other's certificates.
<b>Cryptographic Service Provider (CSP)</b>	Acts as an interface between Microsoft Cryptographic API and private key stores, and performs all cryptographic operations for Microsoft applications and any third-party applications that are properly built on the Windows security framework, such as encrypting and decrypting data, verifying signatures, signing data, and verifying certificates.

Term	Definition
<b>deactivate</b>	Deactivating a user renders them incapable of using Security Manager. This state is reversible; you can reactivate the user later if their certificates have not been revoked.
<b>decrypt</b>	The act of restoring an encrypted file to its original, unprotected state.
<b>decryption private key</b>	<p>The decryption private key decrypts data that has been encrypted with its corresponding encryption public key. Entrust PKI users can encrypt data for a given user, for example Bob, using Bob's encryption public key. Bob is the only user who has access to his decryption private key. Bob uses his decryption private key to decrypt information that has been encrypted for him with his public encryption key.</p> <p>A decryption private key is one half of an asymmetric key pair. The other half is the encryption public key. Asymmetric key pairs are used in Security Manager to encrypt and decrypt the symmetric key. The symmetric key, is used to encrypt and decrypt data, such as documents and email.</p>
<b>desktop user</b>	An Entrust end user who works with an Entrust profile stored on their local computer. Compare with <a href="#">roaming user</a> .
<b>digital signature</b>	Provides a guarantee to a recipient that the signed file came from the person who sent it, and that it was not altered since it was signed. Any other user who has the corresponding verification public key can verify the signature. A digital signature is the result of making a mathematical summary (known as a hash) of data and encrypting the hash using a user's signing private key.
<b>directory</b>	<p>A directory service that contains the names of all Security Manager users and that acts as repository for users' encryption public key certificates.</p> <p>The directory is an online database that updates dynamically—that is, it updates as changes are made to the information it contains. It also contains entries for the Certification Authority, optionally, the Directory Administrator, and the organization itself. The directory also keeps updated lists of revoked certificates (CRLs), and lists of revoked cross-certificates (ARLs).</p>
<b>Directory Information Tree</b>	The logical hierarchical structure of directory information. Entries in the directory are structured hierarchically, using a tree structure.
<b>distinguished name (DN)</b>	The complete name of a directory entry that uniquely identifies a person or entity. DNs of all <a href="#">Security Manager</a> users are stored in the <a href="#">directory</a> .
<b>Document Verifier (DV)</b>	<p>A specialized CA that belongs to a country, It interacts with the CVCA of that country as well as foreign CVCA's. It also interacts with several Inspection Systems of that country. It allows the Inspection Systems to read the biometric information stored on the ePassports of its own country.</p> <p>See also <a href="#">Country Verifying Certification Authority (CVCA)</a>, <a href="#">e-passport</a> and <a href="#">Inspection System</a>.</p>

Term	Definition
DIT	See <a href="#">“Directory Information Tree”</a> .
DN	See <a href="#">“distinguished name (DN)”</a> .
domain	See <a href="#">CA domain</a> .
encrypt	To encrypt a file is to render the file completely unreadable. This means no one, including the owner of the file, can read the file's contents until it is decrypted. Only the owner and the authorized recipients can decrypt the file. The owner determines authorized recipients.
encryption key pair	<p>The encryption key pair comprises an encryption public key and a corresponding decryption private key. Entrust PKI users can encrypt data for a given user, for example Bob, using Bob's encryption public key. Bob is the only user who has access to his decryption private key. Bob uses his decryption private key to decrypt information that has been encrypted for him with his public encryption key.</p> <p>The encryption public key and the decryption private key form the two halves of an asymmetric key pair. Asymmetric key pairs are used in Security Manager to encrypt and decrypt the symmetric key. The symmetric key, is used to encrypt and decrypt data, such as documents and email.</p>
encryption public key	<p>The encryption public key encrypts data that can be decrypted with the decryption private key. Entrust PKI users can encrypt data for a given user, for example Bob, using Bob's encryption public key. Bob is the only user who has access to his decryption private key. Bob uses his decryption private key to decrypt information that has been encrypted for him with his public encryption key.</p> <p>An encryption public key is one half of an asymmetric key pair, the other half is the decryption private key. Asymmetric key pairs are used in Security Manager to encrypt and decrypt the symmetric key. The symmetric key, is used to encrypt and decrypt data, such as documents and email.</p>
end user	Refers to a user of Security Manager client applications.
Entrust certificate file	Contains the necessary information to ensure that files encrypted and signed by someone using an Entrust desktop application in one CA domain can be decrypted and verified by someone using an Entrust desktop application in a non-cross-certified CA domain. All users can export their own certificate file and import any user's certificate file. The filename comprises a user name with a <code>.key</code> filename extension (for example, <code>Alice Gray.key</code> ). See <a href="#">“CA domain” on page 688</a> .
Entrust domain	Comprises everyone in your CA domain and other CAs that your organization is cross-certified with. A CA domain is a group of people (a company, work group or team, educational or governmental institution) who all use Security Manager under the same software license and who certified by the same Certification Authority (CA).

Term	Definition
<b>entrust.ini file</b>	The file that contains important system configuration data that Security Manager client applications (such as <a href="#">Security Manager Administration</a> ) need in order to run. This file is created in the Security Manager root directory on the Security Manager server during installation. You distribute this file along with the appropriate Entrust software to administrators and end users.
<b>Entrust PKI administrator</b>	In Security Manager documentation, refers generally to all predefined administrative roles, including Security Officers, Administrators, Directory Administrators, and Auditors. Also included are any new roles you create to administer Security Manager end users.
<b>Entrust Ready</b>	An Entrust program that applies the Entrust Ready brand to all Entrust and third-party applications that meet Entrust's strict guidelines for compatibility and interoperability with Entrust enhanced security.
<b>e-passport</b>	<p>A passport containing a chip which stores critical information. Also known as a machine readable passport (MRP) or machine readable travel document (MRTD).</p> <p>See also <a href="#">Country Verifying Certification Authority (CVCA)</a>, <a href="#">Document Verifier (DV)</a> and <a href="#">Inspection System</a>.</p>
<b>event manager</b>	Security Manager component that coordinates key management activities such as key updates, DN changes, certificate type changes, update-allowed changes, and move user operations by storing user event status information and sending event messages to clients who request status information; referred to as event manager in this document.
<b>Extended Access Control (EAC)</b>	This is the mechanism used to unlock the biometric data stored in the ePassport chip. EAC ensures that only authorized entities can access the biometric data.
<b>hardware security module (HSM)</b>	A hardware device used to generate key pairs (see <a href="#">key pair</a> ), store the <a href="#">private key</a> , and generate digital signatures.
<b>hash function</b>	<p>A hash function produces a unique value (for example, a series of numbers) when applied to a unique piece of data, such as a document. If even so much as a single letter in the document is altered, the hash function produces a completely different value when applied again to the document.</p> <p>Hash functions are often referred to as one-way functions, meaning that it is extremely difficult to determine the input to the one-way function if you only have the result of applying the function. The result from a hash function is encrypted using the originator's signing private key in order to sign a file.</p>

Term	Definition
<b>hierarchy of CAs</b>	A hierarchical system of multiple CAs in which the <a href="#">Certification Authority (CA)</a> at any level signs the CA certificate of the CAs immediately below it in the hierarchy. Trust derives from a single <a href="#">root CA</a> at the top of the hierarchy. Only the root CA can cross-certify with other CAs.
<b>HSM</b>	See <a href="#">hardware security module (HSM)</a> .
<b>Inspection System</b>	<p>A specialized computer system that is used at a country's ports of entry. Each Inspection System interacts with a DV of its own country. It is used to examine biometric information stored electronically on the ePassports (both domestic and foreign) of people passing through that port.</p> <p>See also <a href="#">Document Verifier (DV)</a> and <a href="#">e-passport</a>.</p>
<b>key</b>	A special number that an encryption algorithm uses to change data, making that data secure.
<b>key history</b>	<p>The collection of decryption private keys belonging to a user. Security Manager stores old keys, the new keys after recovery, and information about the corresponding encryption public keys.</p> <p>If a user loses their keys or forgets their profile password, and needs to have their keys recovered, Security Manager manages this user's collection of decryption private keys, identifying which key is required to decrypt which data.</p>
<b>key lifetime</b>	The length of time a key is valid. All keys have a specific lifetime except the decryption private key which never expires.
<b>key pair</b>	<p>Asymmetric keys come in pairs. <a href="#">Security Manager</a> uses asymmetric keys in both encryption and digital signature operations.</p> <p>In the encryption operation, there is an encryption public key and a decryption private key. The decryption private key decrypts data that has been encrypted with the corresponding encryption public key. Entrust PKI users can encrypt data for a given user, for example Bob, using Bob's encryption public key. Bob is the only user who has access to his decryption private key. Bob uses his decryption private key to decrypt information that has been encrypted for him with his public encryption key.</p> <p>In the digital signature operation, there is a signing private key and a verification public key. The verification public key is used to decrypt a hash value that has been encrypted with the signing private key. An Entrust PKI user, for example Alice, is the only user who has access to Alice's signing private key. Alice uses her signing private key to encrypt the hash value of a file she is signing. Users verify the signature by successfully decrypting the hash value using Alice's verification public key.</p>

Term	Definition
<b>key recovery</b>	<p>The process of generating new activation codes for a user who has lost their profile or has forgotten their password. An Entrust PKI administrator begins the key recovery process and the user completes it, using the new activation codes.</p> <p>When you recover a user's keys, Security Manager sends the user's Entrust desktop application a copy of their encryption key pair history, enabling the user to access previously encrypted data.</p>
<b>key update</b>	<p>The process that replaces old <a href="#">key pair</a> with new ones. During key update, new public key certificates (see <a href="#">certificate</a>) that have no relation to the old keys and certificates are created and users receive new keys and certificates securely.</p>
<b>Lightweight Directory Access Protocol (LDAP)</b>	<p>Lightweight Directory Access Protocol. A directory access protocol (DAP) specified by the Internet Engineering Task Force (IETF).</p>
<b>Master User</b>	<p>An administrator who first helps the site planner to install Security Manager, then maintains the Security Manager service and <a href="#">Security Manager database</a>. The Master User is a highly trusted person in an organization who has a sound working knowledge of the operating system and directories.</p>
<b>multiple authorizations</b>	<p>Certain sensitive operations in Security Manager Administration require multiple authorizations as specified by the organization's security policy that is set in Security Manager Administration. It's a good idea to have at least one more Security Officer than the minimum number required. If one Security Officer forgets their password or cannot be present, the extra Security Officer can provide the final password needed for authorization.</p>
<b>nonrepudiation</b>	<p>Irrefutable evidence that makes it impossible to reject the validity of a signature on a file or transaction.</p> <p>An Entrust digital signature provides nonrepudiation, as well as authentication (guarantees who signed the data) and data integrity (recipients of signed data are alerted if the data has been tampered with).</p>
<b>object class</b>	<p>A directory term. Every entry in the directory belongs to at least one object class. Entries belonging to one object class share similar characteristics (they likely have the same set of attribute types).</p>
<b>object identifier (OID)</b>	<p>An implementation-specific integer or string that uniquely identifies an object. The format of an OID is, for example: 2.6.7.1.47.98.2</p>
<b>organization</b>	<p>A group of people (a company, work group or team, educational or governmental institution) who all use Security Manager under the same software license.</p>



Term	Definition
<b>password check value</b>	The result of using a password that has passed through a hash function numerous times to encrypt a known value. Security Manager only stores the password check value (not the original password).
<b>partitioned CRL</b>	A Certificate Revocation List that is divided into several partitions. Each partition holds up to 750 serial numbers of revoked certificates.
<b>peer-to-peer cross-certification</b>	The process of certifying the trustworthiness of another organization's CA so that users from your CA can validate users from the cross-certified CA. Peer-to-peer cross-certification is ideal between organizations where each organization wants to maintain secure control over its own organization, while forming business relationships.
<b>permissions</b>	In Security Manager, you can create roles with customized permissions to suit your organization's needs. For each role, you can set permissions that specify the administrative operations the role can perform and whether those operations require authorization. For example, your organization may want to create an administrative role with only the permission to add, view, deactivate, and revoke users' certificates.
<b>PKCS</b>	See <a href="#">Public Key Cryptographic Standards (PKCS)</a> .
<b>PKCS #10</b>	<p>Public-Key Cryptography Standards #10, a file format that you generate to initiate an offline trust relationship with another <a href="#">Certification Authority (CA)</a>. The trust relationship can be established either by offline cross-certification or by creating an offline hierarchy of CAs (that is, by installing an offline subordinate CA). A PKCS #10 request contains the CA verification public key and <a href="#">distinguished name (DN)</a> of the CA initiating the request.</p> <p>Also, a validation string is generated at the same time as the request. You must give this validation string (in a secure manner) to the CA with which you want to establish trust, so they can ensure that your PKCS #10 request has not been tampered with.</p>
<b>PKCS #11</b>	An industry standard interface between applications such as the PCU and hardware security modules (see <a href="#">hardware security module (HSM)</a> ). It is supplied by the HSM vendor and exists as a DLL for Windows computers or a Shared Object (.so file) for Solaris computers.
<b>PKCS #12</b>	Part of <a href="#">Public Key Cryptographic Standards (PKCS)</a> , a standard for the format of credentials storage.
<b>PKCS #7</b>	Part of <a href="#">Public Key Cryptographic Standards (PKCS)</a> , a standard that protects data published by RSA Data Security, Inc. PKCS #7 is the security mechanism behind Secure Multipurpose Internet Mail Extensions (S/MIME).
<b>PKI</b>	See <a href="#">public key infrastructure (PKI)</a> .



Term	Definition
<b>PKIX</b>	See <a href="#">public key infrastructure X.509 (PKIX)</a> .
<b>PKIX-CMP subsystem</b>	<p>A secure communication protocol used between Entrust desktop applications and Security Manager. The PKIX-CMP (Public-Key Infrastructure (X.509)-Certificate Management Protocol) subsystem handles requests from all release 5.0 or later Security Manager client applications.</p> <p>It is a part of the Security Manager service.</p>
<b>PostgreSQL</b>	See <a href="#">Security Manager database</a> .
<b>private key</b>	The portion of a <a href="#">key pair</a> that is kept secret by its owner.
<b>profile</b>	<p>An encrypted file containing information about an Entrust PKI user. The Entrust profile contains a user's identity (DN), decryption private keys, their signing keys, and the CA signing keys.</p> <p>The Entrust profile authenticates the user's identity to their CA and allows the user to access their private data. For increased security, store this file permanently on a diskette in a secured location. See also <a href="#">Certification Authority (CA)</a>.</p>
<b>proto-PKIX subsystem</b>	A secure communication protocol used between Entrust connector applications and Security Manager. It handles requests for operations such as key initialization, key update, key expiry, key recovery.
<b>public key</b>	The portion of a key pair that is available in the <a href="#">directory</a> .
<b>Public Key Cryptographic Standards (PKCS)</b>	A set of standard protocols that facilitate the exchange of information, in a secure manner, over the Internet.
<b>public key cryptography</b>	A cryptographic method that uses keys that are public, for encryption and verification, and private, for decryption and digitally signing data.
<b>public key infrastructure (PKI)</b>	<p>1) A system that provides the basis for establishing and maintaining a trustworthy networking environment through the generation and distribution of keys and certificates. See <a href="#">certificate</a>.</p> <p>The Entrust Authority PKI uses Security Manager.</p> <p>2) The foundation technology for providing enhanced Internet security.</p>
<b>public key infrastructure X.509 (PKIX)</b>	A working group within the Internet Engineering Task Force (IETF) that has developed standards for formatting and transporting information within a <a href="#">public key infrastructure (PKI)</a> .
<b>RDN</b>	See " <a href="#">relative distinguished name (RDN)</a> ".

Term	Definition
<b>recover</b>	<p>An operation performed on a user who has a lost or corrupt Entrust profile. Recovering a user's keys generates a new signing key pair and securely retrieves the current encryption public key certificate, decryption private key history, verification public key certificate, and CA verification public key certificate from the Security Manager database.</p> <p>Beginning a key recovery operation generates new activation codes for the user. The user completes the process by entering the new activation codes in the Recover Profile wizard.</p>
<b>reference number</b>	A number (for example, 91480165), obtained from an Security Manager Entrust PKI administrator, which is used along with an authorization code to create a new profile or to recover a lost or corrupt profile. A reference number can only be used once.
<b>registration authority (RA)</b>	The people, processes, and tools used to support the registration of users into the infrastructure. It includes the ongoing administration of users.
<b>relative distinguished name (RDN)</b>	The set of attribute types and distinguished values that uniquely identifies an entry at its level of the directory information tree (DIT).
<b>revocation list distribution point</b>	See <a href="#">“CRL distribution points” on page 690</a> .
<b>revoking user certificates</b>	<p>The process of stopping a user from using Entrust. You must revoke a user's encryption and verification certificates when the user is no longer trusted (for example, if you suspect that their Entrust profile and password compromised by an attacker). You can also revoke certificates even when there is no suspicion of compromise (for example, when a user's DN changes).</p> <p>Depending on which keys you revoke, the user will be unable to log in to Entrust or force a key update during login. Other users, too, may be restricted from encrypting for the user.</p>
<b>roaming user</b>	A user who can encrypt and sign files on computers or kiosks without carrying their digital identity (Entrust profile). You can create roaming users if your organization has the Roaming Server.
<b>role</b>	Your organization can use roles to allow some people to have administrative privileges while restricting other users to an end-user role. Security Manager comes with pre-defined roles that you can customize to suit your organization.
<b>root CA</b>	The <a href="#">Certification Authority (CA)</a> which is at the top of a hierarchy of two or more CAs, which acts as a trust anchor for all CAs in the hierarchy.

Term	Definition
<b>RSA</b>	Rivest Shamir Adleman. A popular public key algorithm, named after its developers Ron Rivest, Adi Shamir, and Leonard Adleman, that can be used for both encryption and digital signatures. See digital signature.  See also <a href="#">RSAPSS</a> .
<b>RSAPSS</b>	PSS stands for Probabilistic Signature Scheme, which is an improved method for creating signatures using the RSA algorithm.  See also <a href="#">RSA</a> .
<b>searchbase</b>	A segment of a larger domain, which allows all users to search more quickly for users and recipients in the same domain or in a cross-certified domain.
<b>Security Manager</b>	An Entrust product that includes an encryption engine, a <a href="#">directory</a> , a database, and administration tools. It is the main part (the server component) of an Entrust PKI system. Security Manager is required as the basis for all managed security solutions from Entrust.  Its primary functions are to create encryption key pairs for users, create certificates for all public keys, manage a secure database of Security Manager information, and enforce an organization's security policies.  Formerly known as Entrust/PKI.
<b>Security Manager Administration</b>	The application in which you administer a Security Manager system. Security Officers, Administrators, and other administrative roles you create can use Security Manager Administration to set the security policy, add users, deactivate users, reactivate users, and so on.
<b>Security Manager Control Command Shell</b>	Security Manager Control Command Shell is a command line utility for Master Users to manage various aspects of Security Manager, such as certificates, the database, and the directory.
<b>Security Manager database</b>	A database that stores information about Security Manager users and the <a href="#">Certification Authority (CA)</a> . The data is encrypted and protected by <a href="#">Master User</a> passwords.
<b>security policy</b>	An organization's security policy governs the use of Security Manager in the organization to achieve security objectives. Included in the security policy are the validity period of certificate revocation lists (CRLs) and the cross-certificate policy.
<b>sensitive operations</b>	Operations that are deemed as requiring authorization. By default, the following operations are considered to be sensitive: setting or changing any aspect of the security policy, adding or deactivating Security Officers or Administrators, changing user properties for Entrust PKI administrators, and cross-certifying with other CAs.
<b>serial number</b>	A unique identifier, such as an employee number, that distinguishes a user in the directory from another user with the same name.

Term	Definition
<b>SHA</b>	Secure hash algorithm. An algorithm developed by NIST and NSA as a hash algorithm to be used with the Digital Signature Standard ("DSS").
<b>signing key pair</b>	The <a href="#">key pair</a> that contains a <a href="#">signing private key</a> and a verification public key.
<b>signing private key</b>	The key that encrypts a hash value that is decrypted with the corresponding verification public key. For example, Alice is the only user who has access to her signing private key, which she uses to encrypt the hash value of a file she is signing, and users verify the signature by successfully decrypting the hash value using Alice's verification public key.
<b>Site Planner</b>	A highly trusted person in an organization who decides and coordinates all aspects of <a href="#">Security Manager</a> installation and setup.
<b>subordinate CA</b>	Any <a href="#">Certification Authority (CA)</a> in a hierarchy that is below the <a href="#">root CA</a> .
<b>superior CA</b>	In a hierarchy of CAs, the superior CA is the one immediately above the current CA in the hierarchy.
<b>symmetric key</b>	A single key that both encrypts and decrypts the same data.
<b>Terminal Authentication</b>	This is one of the steps taken in <a href="#">Extended Access Control (EAC)</a> . It follows <a href="#">Chip Authentication</a> . It is the process of authenticating the <a href="#">Inspection System</a> terminal to the chip in the e-passport. A successful authentication permits the terminal to read the biometric data stored in the e-passport chip.
<b>Terminal Authentication Algorithm</b>	This is the algorithm used by the e-passport infrastructure to authenticate the <a href="#">Inspection System</a> terminal to the e-passport chip.  It is the digital signature algorithm used when the Inspection System computes the response to the challenge presented by the MRTD (see <a href="#">e-passport</a> ). It is also the digital signature algorithm used when generating and verifying the digital signatures on EAC certificates and certificate requests.

Term	Definition
<b>third-party trust</b>	<p>Third-party trust refers to a situation in which two people implicitly trust each other, even though they have not previously established a personal relationship. In this situation, two people can trust each other if they both have a relationship with a common third party, because the third party can vouch for the trustworthiness of the two people.</p> <p>The need for third-party trust is fundamental to any large-scale implementation of a network security product. Public-key cryptography requires access to users' public keys. However, in a large-scale network, it is impractical and unrealistic to expect each user to have previously established relationships with all other users. Plus, because users' public keys must be widely available, the link between a public key and a person must be guaranteed by a trusted third party to prevent masquerading. In effect, users implicitly trust any public key certified by the third party because their organization owns and securely operates the third-party certification agent.</p>
<b>third-party security store</b>	Storage medium for user's Entrust digital ID that is commonly password protected, owned by a third-party vendor, and managed by Entrust .
<b>Triple-DES</b>	A variation of the DES algorithm that uses three 64-bit keys.
<b>user</b>	Any entry in the Security Manager database or directory. A user can be an actual <a href="#">end user</a> or <a href="#">Entrust PKI administrator</a> in your organization, or a non-human entity such as a Web server or other hardware device.
<b>V1 key container</b>	Holds a V1-key-pair and is part of a V1 Entrust digital ID.
<b>V1-key-pair</b>	Key pair that uses only the Client Settings user policies in Security Manager. All versions of Entrust Authority Security Manager can issue V1-key-pairs.
<b>V1 Entrust digital ID</b>	Entrust digital ID that contains one or two V1-key-pairs.
<b>V2 key container</b>	Holds a V2-key-pair and is part of a V2 Entrust digital ID
<b>V2-key-pair</b>	Key pair that uses both the Client Settings and Certificate Definition Settings user policies in Security Manager. If a certificate definition policy setting conflicts with a client policy setting, the certificate definition policy setting is used. Entrust Authority Security Manager 7.0 and later can issue V2-key-pairs.
<b>V2 Entrust digital ID</b>	Entrust digital ID that contains V2-key-pairs. The Entrust digital ID has between one and four key pairs
<b>validation string</b>	A string of alphanumeric characters (for example, 7CN4-YL5D-HP7V) that is automatically generated when you export your Entrust certificate file. Each certificate file has a unique validation string, which you must give to recipients of your certificate file. Use the validation string to confirm that the certificate file someone gives you was not modified since it was created.

Term	Definition
validity period	See <a href="#">key lifetime</a> .
verification public key	The public key portion of a signing key pair used to verify data that has been signed by the corresponding signing private key. The verification public key is stored in a certificate called the verification public key certificate. This certificate is digitally signed by the Certification Authority (CA) to verify that the public key within it is the authentic public key of the identified user.
XML	eXtensible Markup Language. A W3C specification for structured data. Similar to HTML, XML uses tags and attributes to place structured data into text files. XML is different from HTML in that it is a metalanguage and, therefore, does not define specific tags and attributes; it just tells you how to define those tags and attributes.

## A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

- 1-key-pair 114
- 1-Key-Pair User 535
- 1-key-pair users
  - changing from 2-key-pair users 120
  - changing into 2-key-pair users 121
  - creating 119
- 2-key-pair 115
- 2-Key-Pair User 535
- 2-key-pair users
  - changing from 1-key-pair users 121
  - changing into 1-key-pair users 120
- 3-key-pair 116
- 4-key-pair 117

## A

- Acceptable policy OIDs 417
- activating
  - Hardware user type 459
  - Person 4.0 PKI user type 459
- activation codes
  - authorization code 153
  - configuring the lifetime 154
  - definition 153, 687
  - distributing 153
  - managing 153
  - reference number 153
  - reissuing 156
  - reissuing in bulk 328
  - viewing 155
  - viewing expiry dates 155
- adding
  - attributes to directory entries 77
  - attributes to user types 455
  - certificate type description 582
  - directory attribute values 79
  - directory attributes in bulk 323
  - entries to the directory 74
  - existing users 152
  - members to groups 334
  - OIDs to the default policy list 93
  - OIDs to the Security Policy 90

- searchbases 344
- searchbases to Security Manager 347
- searchbases to the directory 345
- subjectAltName component values 260
- user types 452
- users in bulk 288
- Admin Services User Management 535
- Admin Services User Registration 535
- administering
  - attribute certificates 635
  - groups 329
  - roles 353
  - searchbases 341
  - subordinate CAs 509
  - user policies 391
  - users 129
  - users with attribute certificates 650
- Administration Policy 85
- administrative permissions. *See* permissions
- Administrator 354
- Administrator Policy 393
- Algorithm for key pair 440
- Algorithm for profile protection 430
- Allow CA personal addr. book use 414
- Allow personal addr. book use 414
- Allow PKCS#12 Export 421
- Allow S/MIME Secure Receipts 416
- Allow Self Revocation 431
- Allow Spillover File for Tokens 429
- Allow unknown extensions 439
- anchor CVCA permissions 377
- anchor DV permissions 379
- archive files
  - deleting 185
  - moving 185
  - overview 183
  - viewing 185
- archiving
  - archive files 183
  - users 171, 186
- ARL 687
- ASH Policy 393

- ASH Service 355
- ASH subsystem 687
- assigning distinguished names 200
- attribute 687
- attribute certificates
  - administering 635
  - administering users with attribute certificates 650
  - and Security Toolkit for the Java Platform 637
  - creating custom attribute certificate types 639
  - deleting 645
  - examples of using 637
  - issuing 640
  - managing 643
  - overview 636
  - reissuing 643
  - replacing 647
  - saving to a file 648
  - steps for working with attribute certificates 638
  - viewing 648
- attributes
  - adding directory attribute values 79
  - adding to directory entries 77
  - adding to user types 455
  - certificate definition policy attributes 432
  - client policy attributes 407
  - replacing attribute values 80
  - support for international characters 681
- audit logs 687
  - clearing 614
  - field descriptions 612
  - overview 609
  - permissions 372
  - printing 612
  - saving 614
  - sorting 613
  - viewing 609
  - viewing details 611
- Auditor 355
- authentication 687
- authority revocation list 688
- authorization code 153, 688
- authorize 652
- authorizing operations 52
- authorizing operations in bulk files 652
- Auto-Associate MS Outlook 427

## B

- Back up private key 441
- backslash
  - in bulk commands 276
  - in master.certspec 532, 533, 534
- backup 688
- bulk commands
  - arguments 276
  - backslash 276
  - commands 276
  - curly braces 277
  - double quotation marks 277
  - group commands 653
  - reference 651
  - searchbase commands 655
  - syntax 276
  - user commands 658
- bulk files
  - authorizing operations 652
  - creating 279
  - processing 280
  - viewing bulk output log files 283
- bulk operations 314
  - adding directory attributes 323
  - adding users 288
  - advanced processing 285
  - bulk command syntax 276
  - canceling DN changes 313
  - canceling key recovery 299
  - changing distinguished names in bulk 308
  - command reference 651
  - creating bulk files 279
  - creating customized directory entries 296
  - deactivating users 300
  - deleting directory attributes 323
  - deleting users 301
  - notifying client applications 327
  - performing 275
  - permissions 373
  - processing bulk files 280
  - reactivating in bulk 301
  - reissuing bulk operations 328
  - restoring information to the directory 325
  - revoking user certificates in bulk 303
  - setting users for key recovery 298
  - updating key pairs 326
  - viewing bulk output log files 283
- bulk output log files



- failure messages 284
- success messages 283
- viewing 283
- bulk processing 688

## C

- CA 689
  - certificate category 528
  - communication when cross-certifying online 475
  - configuring the Security Policy 84
  - cross-certified CA permissions 375
  - cross-certifying 465
  - deleting an imported CA 209
  - establishing trust with another CA 206
  - exchanging public keys 206
  - exporting CA public keys 207
  - exporting the current CA certificate 102
  - importing public keys 208
  - logging in 216
  - moving users 202
  - permissions 375
  - setting up the CA 83
  - subordinate CA permissions 376
  - viewing CA information 104
  - viewing cross-certificates 495
- CA certificates
  - customizing 591
  - extensions you can exclude 594
- CA directory password 688
- CA domains
  - definition 688
  - using to limit trust in cross-certificates 588
- CA profile 688
- CA signing key pair 688
- CA signing private key 688
- CA verification public key 689
- canceled
  - distinguished name changes 198
  - DN changes in bulk 313
  - key recovery 164
  - key recovery in bulk 299
  - user export operations 213
- CDP. See CRL distributionpoints
- Cert. update date 443
- certificate 689
  - certificate attributes 573
    - defining 576

- editing 575
  - editing a variable 578
- certificate categories 527
  - CA 528
  - configuring options 97
  - cross-certificate 527
  - Enterprise 527
  - permissions 373
  - policy 527
  - Web 527
- certificate definition policies 432
- certificate definitions
  - editing 551
  - excluding subjectAltName 270
  - overview 547
- certificate expiration date 689
- certificate extensions
  - defining 557, 584
  - editing 553, 584
  - excluding 593, 596
  - Extended Key Usage 568
  - extensions you can exclude 594, 595
  - for V1 certificate types 553
  - for V2 certificate types 555
  - in cross-certificates 585
  - overview 547, 581
- Certificate lifetime 432
- certificate revocation 689
- certificate revocation list 689
- certificate revocation list. See CRL
- Certificate Signing Alg 444
- certificate specifications
  - see also master.certspec
  - working with 531
- certificate type description
  - adding 582
  - editing 549
  - for policy certificates 573, 581
  - for user certificates 546
- certificate types
  - associating a certificate extension with a V1 certificate type 554
  - associating a certificate extension with a V2 certificate type 556
  - configuring user certificate types 221
  - creating 572
  - defining replacement certificate types 599
  - extensions for V1 certificate types 553

- extensions for V2 certificate types 555
  - making obsolete 599
  - modifying 572
  - permissions 374
  - predefined 535
  - predefined V2 key-pair types 123
- certificates
  - associating a certificate extension with a V1 certificate type 554
  - associating a certificate extension with a V2 certificate type 556
  - attribute certificates 635
  - categories 525, 527
  - certificate attribute 573
  - certificate definitions 547
  - certificate extensions 547, 581
  - certificate specification examples 569
  - certificate specifications 531
  - certificate type description 546, 573, 581
  - client applications and cross-certificates 469
  - configuring options for certificate categories 97
  - creating a cross-certificate type 580
  - creating certificate types 572
  - creating subordinate CA certificates 510
  - creating user certificate information 545
  - customizing CA certificates 591
  - customizing database fields 601
  - database field example 603
  - defining a certificate extension 584
  - defining certificate attributes 576
  - defining certificate extensions 557
  - defining database fields 602
  - editing a certificate type description 549
  - editing a variable for a certificate attribute 578
  - editing certificate definitions 551
  - editing certificate extensions 553
  - editing variables 565
  - enforcing the use of custom policy certificates 578
  - examples of cross-certificate types 589
  - excluding certificate extensions 593
  - excluding default extensions 596
  - exporting 223
  - exporting subordinate CA certificates 521
  - exporting the current CA certificate 102
  - Extended Key Usage 568
  - extensions for V1 certificate types 553
  - extensions for V2 certificate types 555
  - extensions you can exclude 594, 595
  - issuing a new CRL after revoking 180
  - limiting trust using CA domains 588
  - limiting trust using distinguished names 586
  - limiting trust using policy settings 587
  - making a certificate type obsolete 599
  - mapping policy certificates to certificate definitions 404
  - master.certspec 531
  - modifying a cross-certificate type 580
  - modifying certificate types 572
  - modifying user certificates information 545
  - overview 527
  - permissions 373
  - policy certificates 391, 392
  - predefined certificate types 535
  - protocol certificates 506
  - publishing cross-certificates to the directory 499
  - reasons for revoking 174
  - removing cross-certificates from the directory 497
  - restoring to the directory 201
  - revoking 170, 174, 177
  - revoking a subordinate CA certificate 518
  - revoking cross-certificates 501
  - suspending 171, 181
  - turning off revocation checking 605
  - updating cross-certificates 505
  - V1 certificates 546
  - V2 certificates 546
  - viewing 223
  - working with the master.certspec file 539
- Certification Authority 689
- Certification Authority. See CA
- chaining 523
- chaining directories 523
- changing
  - 1-key-pair users into 2-key-pair users 121
  - 2-key-pair users into 1-key-pair users 120
  - distinguished names 193
  - distinguished names in bulk 308
  - format of key identifiers 471
  - password 51
  - profiles 243
  - user properties in bulk 314
  - V2 users to V1 users 247
- changing DNSs 690
- Check e-mail on encryption 427
- Check e-mail on verification 427
- checking permission dependencies 368
- Chip Authentication 690

## ■ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ■

- clearing audit logs 614
- client applications
  - definition 690
  - notifying 242
  - notifying in bulk 327
- client policies 407
- Code Signing 538
- combined CRL 690
- commands. *See* bulk commands
- configuring
  - auto-population of the subjectAltName from the directory 257
  - directory search preferences 59
  - encryption OIDs for users 237
  - general preferences 57
  - general user properties 219
  - group information preferences 64
  - key pairs 107
  - key-pair users 118
  - lifetime of activation codes 154
  - OIDs for users 237
  - OIDs in the Security Policy 90
  - options for certificate categories 97
  - queued requests 95
  - search performance preferences 60
  - Security Manager Administration preferences 57
  - Security Manager license information 55
  - Security Policy 84
  - subjectAltName values 249
  - user certificate types 221
  - user key updates 230
  - user list preferences 62
  - user properties 219
  - V1 1-key-pair users 119
  - V2 key-pair users 122
  - verification OIDs for users 237
- Content Scanner SMTP 426
- conventions 25
- converting V2 users to V1 users 247
- copying
  - roles 361
  - user policies 398
- Country Verifying Certification Authority 690
- creating
  - 1-key-pair users 119
  - bulk files 279
  - cross-certificate type 580
  - custom attribute certificate types 639
  - customized directory entries in bulk 296
  - debug log 615
  - groups 332
  - master.certspec 541
  - new users 146
  - profiles 42, 158
  - reports 619
  - roles 361
  - subordinate CA certificates 510
  - user certificate information 545
  - user policies 398
  - users 203
- creating certificate types 572
- CRL 689
- CRL distribution points 525, 690
- CRL grace percentage 428
- CRL grace period 428
- CRL settings in the Security Policy 88
- CRL. *See* certificate revocation lists
- Cross Cert DNs 427
- cross-certificates 690
  - applications of certificate extensions 585
  - certificate category 527
  - creating a cross-certificate type 580
  - customizing 580
  - examples of cross-certificate types 589
  - extensions you can exclude 595
  - limiting trust using CA domains 588
  - limiting trust using distinguished names 586
  - limiting trust using policy settings 587
  - modifying a cross-certificate type 580
  - setting policy constraints requirements in protocol certificates 506
  - taking off hold 504
  - viewing 495
- cross-certification 690
  - see also* cross-certifying
- cross-certified CAs
  - permissions 375
- cross-certifying
  - client applications and cross-certificates 469
  - key identifiers 470
  - methods 473
  - offline 484
  - online 474
  - overview 467
  - performing offline 484
  - performing online 475

- with other CAs 465
- Cryptographic Service Provider 690
- CSP 690
- CSP to manage keys 442
- curly braces 277
- customer support 28
- customizing
  - CA certificates 591
  - certificates 525
  - cross-certificates 580
  - database fields 601
  - Enterprise certificates 544
  - Web certificates 544
- customizing policy certificates 571
- CVCA 690
- CVCA permissions 380

## D

- database fields
  - customizing 601
  - defining 602
  - examples 603
- deactivate 691
- deactivating
  - users 169, 172
  - users in bulk 300
- debug log 615
- decrypt 691
- decryption private key 691
- Default
  - Enterprise certificate type 535
  - Web certificate type 538
- default. See predefined
- defining
  - certificate extensions 557, 584
  - database fields 602
  - replacement certificate types 599
- Delay single login 424
- deleting
  - archive files 185
  - attribute certificates 645
  - attribute values 81
  - directory attributes 81
  - directory attributes in bulk 323
  - entries in the directory 76
  - groups 339
  - OIDs from the Security Policy 92
  - roles 370
  - searchbases 351
  - see also removing
  - subjectAltName component values 260
  - user policies 405
  - users from the directory in bulk 301
- deleting attribute values 81
- Desktop Admin 535
- desktop user 691
- digital signature 691
- directory 523, 691
  - adding attribute values 79
  - adding attributes in bulk 323
  - adding attributes to directory entries 77
  - adding entries 74
  - adding searchbases 345
  - deleting attributes 81
  - deleting attributes in bulk 323
  - deleting entries 76
  - finding entries 72
  - issues with international characters 682
  - permissions 376
  - preparing 523
  - publishing cross-certificates 499
  - referring 523
  - removing cross-certificates 497
  - replacing attribute values 80
  - restoring information in bulk 325
- Directory Administrator
  - changing password 82
  - role 355
- Directory Browser
  - adding attribute values 79
  - adding attributes to directory entries 77
  - adding entries to the directory 74
  - changing the Directory Administrator password 82
  - deleting attribute values 81
  - deleting directory attributes 81
  - deleting entries in the directory 76
  - finding entries in the directory 72
  - locking 71
  - logging in 70
  - overview 69
  - replacing directory attribute values 80
- Directory Information Tree 691
- Disable single login 409
- distinguished name 691
- distinguished names

# ■ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ■

- assigning 200
- canceling changes 198
- canceling DN changes in bulk 313
- changing 193
- changing in bulk 308
- definition 192
- entering international characters 679
- including email addresses 132
- modifying 192
- troubleshooting changes 197
- using to limit trust in cross-certificates 586
- viewing user history 237
- distributing activation codes 153
- DIT 691
- DN 691
- DN encoding formats 414
- DN. See distinguished name
- Do not process anyPolicy 420
- Do not process policy mappings 418
- Document Signer Policy 393
- Document Verifier 691
- documentation
  - conventions 25
  - obtaining 27
  - providing feedback 27
  - related documentation 26
  - revision information 24
- domain 692
- Domain Controller 538
- double quotation marks 277
- Dual Usage No Key Backup Policy 393
- Dual Usage Policy 393
- dual-usage key pair 108
- DV 691
- DV permissions 377

## **E**

- EAC 693
- EAC Administrator 355
- EAC Auditor 356
- EAC DV CKM Administrator 356
- EAC Self-Service 356
- editing
  - a variable for a certificate attribute 578
  - certificate attributes 575
  - certificate definitions 551
  - certificate extensions 553, 584
  - certificate type description 549
  - certificate variables 565
- EFS encryption key pair 111
- EFS Policy 393
- EFS User 535
- EKU. See Extended Key Usage
- email addresses
  - in Security Manager 132
  - including in distinguished names 132
- Email Content Scanner 535
- Enable CAPI Synchronization 422
- Enable cert. update date 443
- Enable CMP override 438
- Enable the use of a Cert cache 425
- Enable the use of a CRL cache 425
- Enable the use of a XCert cache 425
- Enable the use of an ARL cache 425
- encrypt 692
- encryption key pair 109, 692
- encryption OIDs
  - configuring for users 237
  - configuring in the Security Policy 90
- Encryption Policy 393
- encryption public key 692
- Encryption\_p10 Policy 393
- End User 355
- end user 692
- End User Policy 393
- Enforce client policy 439
- Enforce identity usage 414
- Enforce protected key transfer 429
- Enforce S/MIME 416
- Enforce token usage 413
- Enterprise
  - certificate category 527
  - customizing certificates 544
  - predefined certificate types 535
- Enterprise Domain Controller 536
- Enterprise Domain Controller Policy 393
- Enterprise Machine 536
- Enterprise Machine Policy 393
- Entrust Authority Security Manager Administration. See Security Manager Administration
- Entrust certificate file 692
- Entrust domain 692
- Entrust PKI administrator 693
- Entrust Ready 693
- entrust.ini 693

- e-passport 693
- ePassport - Document Signer 536
- ePassport - IS Attached Client 536
- ePassport - IS Concentrator 536
- ePassport - IS Standalone Client 536
- ePassport - Master List Signer 536
- ePassport - Master List Signer Administrator 536
- ePassport - SPOC Administrator 536
- ePassport - SPOC Client 536
- ePassport - SPOC DV Client 536
- ePassport - SPOC Server 536
- event manager 693
- Exclude basicConstraints 436
- Exclude CDP 436
- Exclude certificatePolicy 437
- Exclude entrustVersInfo 436
- Exclude privateKeyUsagePeriod 436
- Exclude subjectAltName 437
- Exclude subjectAltName parts 437
- Exclude subjectKeyId 437
- excluding
  - certificate extensions 593, 596
  - subjectAltName from certificate definitions 270
- Export 537
- exporting
  - CA public keys 207
  - canceling user export operations 213
  - current CA certificate 102
  - subordinate CA certificates 521
  - user certificates 223
  - user policies 406
  - user template file 451
  - users 210, 217
- exporting subjectAltName component values 267
- Express Search Source Order 426
- Extended Access Control 693
  - CVCA permissions 376
  - DV permissions 379
- Extended Key Usage 568

## F

- finding
  - entries in the directory 72
  - users 134
  - users by directory attributes 142
  - users by Entrust properties 134
- Force client key generation in CSP 443

- Force Original CD Compliance 416
- foreign CVCA permissions 378
- FPKI compliance 419

## G

- Generate key at client 441
- glossary 687
- groups
  - adding members 334
  - administering 329
  - bulk commands 653
  - creating 332
  - deleting 339
  - overview 329
  - permissions 382
  - removing members 336
  - renaming 338
  - viewing 330

## H

- Hardware
  - activating the user type 459
  - user type 449
- hardware security module 693
- hash functions 693
- hierarchies
  - definition 509
- hierarchy of CAs 694
- HSM 693, 694
- HTTP Proxy for CRL Requests 420

## I

- ICE settings ignored 425
- ICE settings signed 425
- Ignore per user lifetime 434
- import files
  - contents 210
  - viewing 212
- importing
  - CA public keys 208
  - user policies 406
  - user template file 461
  - users 215
- initial.certspec 531
  - see also master.certspec

## ■ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ■

- Inspection System 694
- Inspection System permissions 381
- installing
  - online help 36
  - Security Manager Administration 32
- international characters
  - attribute support 681
  - directory issues 682
  - displayed in Security Manager Administration 685
  - entering from an ASCII keyboard 683
  - entering in Security Manager Administration 683
  - entering using Microsoft Input Method Editor 684
  - in distinguished names 679
  - when to use 681
- IPSec Device 537
- IS. See Inspection System
- issuing
  - attribute certificates 640
- issuing a new CRL after revoking a certificate 180

## K

- key 694
- Key can sign CMP 439
- key expiry dates 230
- key history 694
- key identifiers
  - changing the format 471
  - in cross-certification 470
- key lifetime 694
- key lifetimes 230
- key pair 694
- key pairs
  - 1-key-pair 114
  - 2-key-pair 115
  - 3-key-pair 116
  - 4-key-pair 117
  - configuring 107
  - configuring key-pair users 118
  - dual-usage key pair 108
  - EFS encryption key pair 111
  - encryption key pair 109
  - models 113
  - nonrepudiation key pair 110
  - overview 108
  - recovering 162
  - signing key pair 110
  - supported by Security Manager 113

- updating 240
  - V1 114
  - V2 114
- key recovery 695
  - canceling 164
  - setting users 163
- Key type for encryption 412
- Key type for signatures 411
- key update 695
- key updates
  - configuring 230
- Key usage policy 441

## L

- LDAP 695
- license information
  - configuring 55
  - permissions 382
- Lightweight Directory Access Protocol 695
- locking
  - Directory Browser 71
  - Security Manager Administration 49
- log files
  - see also audit logs
  - viewing 608, 609
- logging in
  - Directory Browser 70
  - Security Manager Administration 46
  - to the new CA 216
- Login attempt window 425
- login messages received for revoked certificates 175
- Login timeout (minutes) 409

## M

- Management Client 416
- managing
  - activation codes 153
  - attribute certificates 643
- Master List Signer Policy 393
- Master User 695
- master.certspec 531
  - contents 532
  - creating 541
  - examples 569
  - navigating 542
  - opening 541

- processing changes 543
- working with the file 539
- Maximum bad login attempts 424
- Message in Entrust-Ready clients 412
- Messaging Server 537
- Messaging Server SMTP 430
- Minimum PKCS#12 Hash Count 421
- modifying
  - certificate types 572
  - cross-certificate type 580
  - distinguished names 192
  - roles 363
  - searchbases 349
  - subjectAltName component values 260
  - user certificate information 545
  - user policies 402
  - user template 447
  - user types 447
- moving
  - archive files 185
  - decryption private keys 203
  - user certificates 204
  - users 203
- MS CMP Signing Policy 394
- MS EFS Policy 394
- MS VPN Client Machine 538
- MS VPN Client User 537
- MS VPN Server 538
- multiple authorizations 695

## N

- navigating the master.certspec file 542
- nonrepudiation 695
- nonrepudiation key pair 110
- Nonrepudiation Policy 394
- Nonrepudiation User 537
- Nonrepudiation/EFs User 537
- notifying client applications 242
- notifying client applications in bulk 327
- Number of key pairs 423

## O

- object class 695
- object identifier 695
- obsolete certificate types 599
- obtaining

- documentation 27
- technical assistance 28
- offline cross-certification
  - overview 484
  - performing 484
  - requirements 484
- OIDs
  - adding to the default policy list 93
  - adding to the Security Policy 90
  - configuring for users 237
  - configuring in the Security Policy 90
  - definition 695
  - deleting from the Security Policy 92
  - permissions 382
  - removing from the default policy list 94
- online cross-certification
  - communication between CAs 475
  - overview 474
  - performing 475
  - requirements 474
- online help
  - installing 36
  - uninstalling 38
  - upgrading 37
- Only latest key can sign CMP 439
- opening the master.certspec file 541
- organization 695
- Organizational Unit 449

## P

- partitioned CRL 696
- password check value 696
- Password expires in (weeks) 407
- Password history 407
- Password length (characters) 408
- Password needs lowercase letter 408
- Password needs non-alpha char. 408
- Password needs number 409
- Password needs uppercase letter 408
- passwords
  - changing 51
  - changing the Directory Administrator password 82
- peer-to-peer cross-certification 696
- Perform dir. consistency check 415
- performing
  - bulk operations 275
  - offline cross-certification 484



# ■ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ■

- online cross-certification 475
- permissions 696
  - anchor CVCA 377
  - anchor DV 379
  - audit log 372
  - bulk operations 373
  - CA permissions 375
  - certificate categories 373
  - certificate types 374
  - certificates 373
  - checking dependencies 368
  - cross-certified CA permissions 375
  - CVCA 380
  - CVCA permissions 376
  - directory 376
  - DV 377
  - DV permissions 379
  - Extended Access Control permissions 376, 379
  - foreign CVCA 378
  - groups 382
  - Inspection Systems 381
  - license information 382
  - overview 371
  - policy OIDs 382
  - queued requests 383
  - reference 371
  - reports 373
  - roles 384
  - searchbases 384
  - security policy 385
  - subordinate CA 376
  - user templates 386
  - users 386
- Permit desktop 413
- Permit roaming 413
- Permit Server Login usage 414
- Person 449
- Person 4.0 PKI
  - activating 459
  - user type 449
- PKCS 696, 697
- PKCS #10 696
- PKCS #11 696
- PKCS #12 696
- PKCS #7 696
- PKCS10 2-Key-Pair User 537
- PKI 696, 697
- PKIX 697
- PKIX compliance 418
- PKIX-CMP subsystem 697
- polcert\_certdefn. See certificate definition policies
- polcert\_cliset. See client policies
- Policy Certificate expires in (days) 407, 432
- policy certificates
  - category 527
  - customizing 571
  - enforcing 578
  - mapping to certificate definitions 404
  - see also user policies
  - viewing 397
- PostgreSQL 697
- predefined
  - certificate types 535
  - Enterprise certificate types 535
  - user policies 393
  - Web certificate types 538
- preferences
  - configuring 57
  - directory searches 59
  - general preferences 57
  - group information 64
  - search performance 60
  - user lists 62
- preparing the directory 523
- Prevent single login register 424
- printing audit logs 612
- private key 697
- Private key export from CAPI? 423
- Private key export from CSP 442
- Private key usage period 433
- processing
  - bulk files 280
  - changes to master.certspec 543
- professional services 28
- profile export
  - allowing 244
  - overview 245
- profiles
  - allowing profile export 244
  - changing 243
  - creating 42, 158
  - definition 697
  - recovering 66, 165, 217
- Protect key storage for CSP 442
- proto-PKIX subsystem 697
- providing feedback on documentation 27

- public key 697
- Public Key Cryptographic Standards 697
- public key cryptography 697
- public key infrastructure 697
- public key infrastructure X.509 697
- Public Token Certs 429
- Publish at key update 435
- Publish expired certs 435
- Publish revoked certs. 434
- publishing
  - cross-certificates to the directory 499
- Publishing DN 435
- Publishing policy 434

## Q

- queued requests
  - configuring 95
  - permissions 383

## R

- RDN 698
- reactivating
  - users 173
  - users in bulk 301
- recover 698
- recovering
  - profiles 66, 165, 217
  - user key pairs 162
- reference number 153, 698
- referring directories 523
- Reg. Client type 428
- Reg. Pwd Max Fail 428
- reissuing
  - activation codes 156
  - attribute certificates 643
- related documentation 26
- relative distinguished name 698
- removing
  - cross-certificates from the directory 497
  - members from groups 336
  - OIDs from the default policy list 94
  - see also deleting
  - users 171
  - users from the database 191
- renaming groups 338
- replacing

- attribute certificates 647
- attribute values 80
- reports
  - contents 616
  - creating 619
  - fields 629
  - formats 617
  - permissions 373
- Require policy 418
- restoring
  - information in bulk 325
  - user certificates to the directory 201
- restricting users 169
- retrieving archived users 186
- revision information 24
- revocation list distribution point 698
- Revoke superseded certs. 444
- revoking
  - certificates 177
  - cross-certificates 501
  - issuing a new CRL 180
  - reasons 174
  - subordinate CA certificate 518
  - user certificates 170, 174, 698
  - user certificates in bulk 303
- Roaming Server 537
- roaming user 698
- role 698
- roles
  - administering 353
  - checking permission dependencies 368
  - copying 361
  - creating 361
  - deleting 370
  - modifying 363
  - overview 353
  - permissions 371, 384
  - predefined 354
  - viewing 358
- root CA 698
  - in a hierarchy 509
- RSA 699
- RSAPSS 699

## S

- saving
  - attribute certificates to a file 648

# ■ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ■

- audit logs 614
- searchbase 699
- Searchbase Search Order 427
- searchbases
  - adding 344
  - adding to Security Manager 347
  - adding to the directory 345
  - administering 341
  - bulk commands 655
  - deleting 351
  - modifying 349
  - overview 341
  - permissions 384
  - viewing 342
- Secure Delivery SMTP 426
- Security Manager
  - adding searchbases 347
  - configuring license information 55
  - definition 699
  - email addresses 132
  - license information 55
  - predefined roles 354
- Security Manager Administration 699
  - configuring preferences 57
  - displaying international characters 685
  - entering international characters 683
  - getting started 39
  - hiding the toolbar 50
  - installing 32
  - locking 49
  - log files 608
  - logging in 46
  - overview 39
  - reports 616
  - showing the toolbar 50
  - uninstalling 35
  - upgrading 34
  - using tokens 40
  - using with multiple Security Managers 44
- Security Manager Administration Online Help. See online help
- Security Manager Control Command Shell 699
- Security Manager database 699
- Security Manager Directory Administrator
  - see Directory Administrator
- Security Officer 354
- Security Officer Policy 394
- Security Policy
  - adding OIDs 90
  - adding OIDs to the default policy list 93
  - Administration Policy 85
  - configuring 84
  - deleting OIDs 92
  - encryption OIDs 90
  - interaction between CRL settings 88
  - permissions 385
  - removing OIDs from the default policy list 94
  - verification OIDs 90
- security policy 699
- Security Toolkit for the Java Platform 637
- Self-admin policy 429
- Self-Administration Server Administrator 356
- sensitive operations 699
- serial number 699
- Server Login 356
- Server Login Policy 394
- setting
  - criticality of the subjectAltName extension 273
  - policy constraints requirements in protocol certificates 506
  - users for key recovery in bulk 298
- setting users for key recovery 163
- SHA 700
- signing key pair 110, 700
- signing private key 700
- Site Planner 700
- Smart Card Logon for MS Security Framework Users 537
- Smart Card Logon for PKCS#11 Users 537
- sorting audit logs 613
- special characters in user names 133
- SPOC Administrator 356
- SPOC Administrator Policy 394
- SPOC Role 357
- SPOC Self-Service Role 357
- SPOC Server Login Policy 394
- Standalone EFS User 537
- state. See user states
- SubjectAltName
  - criticality 438
- subjectAltName
  - adding component values 260
  - components 251
  - configuring auto-population from the directory 257
  - deleting component values 260
  - excluding from certificate definitions 270
  - exporting component values 267

- modifying component values 260
- overview 250
- setting the criticality 273
- updating component values from the directory 263
- using 250
- values 249
- viewing component values 267
- subordinate CA 700
- subordinate CA certificate
  - taking off hold 520
- subordinate CA certificates
  - creating 510
  - exporting 521
- subordinate CAs
  - administering 509
  - creating subordinate CA certificates 510
  - exporting subordinate CA certificates 521
  - permissions 376
  - revoking a subordinate CA certificate 518
- superior CA 700
- support 28
- suspending user certificates 171, 181
- Symmetric encryption algorithms 410
- symmetric key 700
- Symmetric Key Access 430, 442

## T

- taking a subordinate CA certificate off hold 520
- taking cross-certificates off hold 504
- Tcl 276
  - additional resources 278
  - output 278
- technical support 28
- template definition file. See user template file
- Terminal Authentication 700
- Terminal Authentication Algorithm 700
- terms 687
- testing the user template file 462
- third-party security store 701
- third-party trust 701
- Timestamping Agent 537
- Timestamping Agent Critical 537
- tokens
  - drivers for readers 41
  - supported 40
- toolbar
  - hiding 50

- showing 50
- Triple-DES 701
- Truepass Server 538
- TruePass Server Multidomain Primary 538
- TruePass Server Verification Policy 394
- turning off revocation list checking 605
- typographic conventions 25

## U

- uninstalling
  - online help 38
  - Security Manager Administration 35
- Unprotected CAPI key storage? 422
- Update cert. at % of lifetime 443
- updating
  - cross-certificate 505
  - key pairs 240
  - key pairs in bulk 326
  - subjectAltName component values from the directory 263
- upgrading
  - online help 37
  - Security Manager Administration 34
- Use CMP publish flag 439
- User Administrator 357
- user policies
  - administering 391
  - certificate definition policies 432
  - client policies 407
  - copying 398
  - creating 398
  - deleting 405
  - exporting 406
  - importing 406
  - mapping policy certificates to certificate definitions 404
  - modifying 402
  - overview 392
  - predefined 393
  - viewing 395
  - viewing policy certificates 397
- User Reg Service (Admin Services) 357
- user states 131
- user template file
  - exporting 451
  - importing 461
  - modifying 447
  - overview 448

## ■ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ■

- permissions 386
- testing 462
- user types
  - adding 452
  - adding attributes 455
  - available user types 449
  - Hardware 449
  - importing the user template file 461
  - modifying 447
  - Organizational Unit 449
  - overview 448
  - Person 449
  - Person 4.0 PKI 449
  - purpose 448
  - testing user template file 462
  - Web server 449
- users
  - adding 152
  - adding in bulk 288
  - administering 129
  - administering with attribute certificates 650
  - allowing profile export 244
  - archive files 183
  - archiving 171, 186
  - assigning distinguished names 200
  - bulk commands 658
  - canceling DN changes 198
  - canceling key recovery 164
  - canceling key recovery in bulk 299
  - canceling user export operations 213
  - changing distinguished names 193
  - changing profiles 243
  - changing user properties in bulk 314
  - configuring certificate types 221
  - configuring encryption OIDs for users 237
  - configuring general properties 219
  - configuring key updates 230
  - configuring the lifetime of activation codes 154
  - configuring user properties 219
  - configuring verification OIDs 237
  - converting from V2 to V1 247
  - creating 146, 203
  - creating profiles 158
  - deactivating 169, 172
  - deactivating in bulk 300
  - definition 129
  - deleting in bulk 301
  - distributing activation codes 153

- email addresses in Security Manager 132
- exporting 210, 217
- exporting certificates 223
- finding 134
- finding by directory attributes 142
- finding by Entrust properties 134
- importing 215
- including email addresses in distinguished names 132
- managing activation codes 153
- modifying distinguished names 192
- moving 203
- moving to a new CA 202
- notifying client applications 242
- permissions 386
- reactivating 173
- reactivating in bulk 301
- reasons for revoking certificates 174
- recovering profiles 165, 217
- recovering user key pairs 162
- reissuing activation codes 156
- reissuing activation codes in bulk 328
- removing 171
- removing from the database 191
- restricting 169
- retrieving 186
- revoking user certificates 170, 174
- revoking user certificates in bulk 303
- setting for key recovery 163
- setting for key recovery in bulk 298
- states 131
- suspending user certificates 171, 181
- troubleshooting DN changes 197
- updating key pairs in bulk 326
- using special characters in user names 133
- viewing activation codes and expiry dates 155
- viewing certificates 223
- viewing DN change history 237
- viewing import files 212
- using special characters in user names 133
- using Tcl 276

## V

- V1
  - 1-key-pair users 119
  - certificates 546
  - client applications 114, 123
  - Entrust digital ID 701

■ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ■

- key container 701
- key pairs 114
  - V1-key-pair 701
- V1 users 247
- V2
  - certificates 546
  - client applications 114, 126
  - Entrust digital ID 701
  - key container 701
  - key pairs 114
  - key-pair certificate types 123
  - key-pair users 122
  - V2-key-pair 701
- V2 users 247
- validation string 701
- verification OIDs
  - configuring for users 237
  - configuring in the Security Policy 90
- Verification Policy 394
- verification public key 702
- Verification\_p10 Policy 394
- viewing
  - activation codes 155
  - archive files 185
  - audit log details 611
  - audit logs 609
  - bulk output log files 283
  - CA information 104
  - contents of attribute certificates 648
  - cross-certificates 495
  - DN change history 237
  - expiry dates of activation codes 155
  - groups 330
  - import files 212
  - roles 358
  - searchbases 342
  - Security Manager Administration log files 608
  - subjectAltName component values 267
  - user certificates 223
  - user policies 395

Web server 449

## X

- X.509 528
- XAP Server 538
- XML 702

## W

- Web
  - certificate category 527
  - customizing certificates 544
  - predefined certificate types 538
- Web Server 538

■ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ■

■ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ■



■ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ■

■ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ■