

Security Platform Operations Guide

Manual

Notice

This document contains proprietary and confidential material of Comverse, Inc. This document is furnished under and governed by either a license or confidentiality agreement. Any unauthorized reproduction, use, or disclosure of this material, or any part thereof, is strictly prohibited.

The material furnished in this document is believed to be accurate and reliable. However, no responsibility is assumed by Comverse, Inc. for the use of this material. Comverse, Inc. reserves the right to make changes to the material at any time and without notice. This document is intended for information and operational purposes only. No part of this document shall constitute any contractual commitment by Comverse, Inc.

© 2010 Comverse, Inc. All rights reserved.

Portions of this documentation and of the software herein described are used by permission of their copyright owners.

Comverse, its logo, the spark design, and Netcentrex are registered trademarks of Comverse Technology, Inc. or its subsidiaries in the United States and may also be registered in other countries.

Other denoted product names of Comverse or other companies may be trademarks or registered trademarks of Comverse, Inc. or its subsidiaries, or their respective owners. Portions of the software may be subject to copyrights owned by Infor Global Solutions (Michigan), Inc.

Corporate Headquarters
200 Quannapowitt Parkway
Wakefield, MA 01880 USA
Tel: (781) 246-9000
Fax: (781) 224-8143
www.comverse.com

Revision History

The following table lists the document changes since the initial publication.

Date	Chapter/ Appendix	Description
10/15/2010		Initial publication for the 3.5 50 release.

Contents

Revision History.....	iii
Figures	xi
Tables.....	xiii
Notational Conventions.....	xv
Comverse ONE Documentation List	xvii

Chapter 1 Introduction 1

Welcome	3
New Features for This Release.....	3
Who Should Use This Document	3
Organization of This Document.....	3

Chapter 2 Security Overview 5

Security Solution Overview	7
Identity Management and Authentication	7
Identity Management Framework	8
Identity Management API	8
Policy Management and Authorization.....	9
Policy Management Framework.....	9
Policy Management API	9
Audit Management and Accountability	10
Audit Management Framework	10
Audit Management API	10
Data Encryption and Credentials Management.....	10
Data Encryption and Credentials Management Framework.....	10
Data Encryption and Credentials Management APIs	11

Chapter 3 Identity Management 13

Identity Management Overview.....	15
Getting Started with Identity Management	15
Identity Management Interface	15
Identity Management Tasks	16
Configuration	16
Understanding and Managing Security Realms and Realm Groups	16
Understanding and Managing Security Roles	22
Understanding and Managing Password Policies	24
Working with User Accounts	24
User Account Management.....	24
Password Management.....	30
Bulk Account Management Operations	31

Chapter 4 Policy Management..... 33

Policy Management Overview	35
----------------------------------	----

What Is XACML?	35
XACML Policies	35
Policy Enforcement Point	36
Policy Decision Point	36
Policy Administration Point	37
Getting Started with Policy Management	37
Policy Management Interface	37
Policy Management Tasks	37
Working with Rules	38
Rule Management	38
Bulk Rule Operations	40
Working with Policies	41
Policy Management	41
Bulk Policy Operations	45
Manual Policy Creation	46
Chapter 5 Audit Management	47
Audit Management Overview	49
Getting Started with Audit Management	49
Audit Management Interface	49
Audit Management Tasks	50
Audit Management Operations	50
Enabling/Disabling Auditing	50
Querying Audit Records	50
Purging Audit Records	51
Audit Record Format	51
Header	51
Originator	52
Initiator	52
Target	52
Source	53
Event	53
Audit Event Index	53
XDAS Event Codes	53
XDAS Event Outcome Codes	57
Chapter 6 Encryption Key and Credentials Management	61
Encryption Key and Credentials Management Overview	63
Getting Started with Key/Credentials Management	64
Key and Credentials Management Interface	64
Key and Credentials Management Tasks	64
Symmetric Key Management Operations	65
Viewing Symmetric Keys	65
Creating Symmetric Keys	65
Disabling Symmetric Keys	65
Enabling Symmetric Keys	66
Deleting Symmetric Keys	66
Credentials Management Operations	66
Viewing Credentials	66

Creating Credentials	67
Modifying Credentials	68
Deleting Credentials	68
Publishing Database Credentials	69
Bulk Credentials Operations	70
Operating System Credentials	71
Upgrades and Database/OS Credentials	71
Changing Database Passwords after an Upgrade	71
Changing OS User Passwords after an Upgrade	72

Chapter 7 Security GUI..... 73

Security GUI	75
Logging In to the Security GUI	75
Logging Out of the Security GUI	77
Identity Management	77
Working with User Accounts	78
Working with Security Realm Groups	84
Working with Security Roles	90
Working with Security Realms	91
Key Management	99
Viewing Keys	99
Creating Keys	99
Disabling/Enabling Keys	100
Deleting Keys	100
Policy Management	101
Working with Policies	101
Working with Rules	106
Audit Management	110
Credentials Management	111
Viewing Database Credentials	111
Deleting Database Credentials	112
Creating Database Credentials	112
Publishing Database Credentials	113
Viewing Network Credentials	114
Deleting Network Credentials	114
Creating Network Credentials	114

Appendix A Security-Related Management Shell Commands 117

Overview	119
Global Options	119
General Command Syntax	119
Identity Management Commands	120
add_group	120
add_realm	120
add_role	121
add_user	122
change_password	123
disable_user	123
enable_user	123

find_users	123
list_attributes	123
list_groups	124
list_purged_users	124
list_realms	124
list_roles	124
list_users	125
lock_user	125
modify_group	125
modify_realm	126
modify_user	127
remove_group	128
remove_realm	128
remove_role	128
remove_user	129
reset_password	129
unlock_user	129
Policy Management Commands	129
create_auth_policy	129
create_auth_rule	130
list_auth_policy	130
list_auth_rule	131
modify_auth_policy	131
modify_auth_rule	131
publish_policy	132
remove_auth_policy	132
remove_auth_rule	132
resync_policy	133
Audit Management Commands	133
build_report	133
Key Management Commands	134
create_key	134
delete_key	134
disable_key	134
enable_key	134
list_keys	135
Credentials Management Commands	135
list_credential	135
publish_credential	135
remove_credential	136
store_credential	136

Chapter 8 Database Reference..... 139

Overview	141
SEC_AA_EVENT	144
SEC_DPM_PASSWORD	146
SEC_IDM_COUNTER	147
SEC_IDM_GROUP	148
SEC_IDM_GROUP_ATTRIBUTE	149
SEC_IDM_GROUP_ROLE	150

SEC_IDM_PURGED_USERS	151
SEC_IDM_REALM	152
SEC_IDM_REALM_ATTRIBUTE	153
SEC_IDM_ROLE	154
SEC_IDM_USER	155
SEC_IDM_USER_ATTRIBUTE	157
SEC_IDM_USER_GROUP	158
SEC_IDM_USER_SESSION	159
SEC_KM_KEYS	160
SEC_KM_SERVER_KEYS	161
SEC_NETWORK_DEV_CRED	162
SEC_PASS_POLICY	163
SEC_PASS_POLICY_HIST	165
SEC_PM_POLICY	166
SEC_PM_RULE	167
SEC_PM_RULE_PM_POLICY	168
SEC_SESS_POLICY	169
 Appendix B Attribute Conflict-Resolution Rules	 171
Overview	173
Conflict-Resolution Rules	173
Examples	173
Scenario 1	173
Scenario 2	174
Scenario 3	174
Scenario 4	174
 Appendix C Security Server Database Restore Operations	 177
Overview	179
Automated Security Data Export/Database Backup	179
Methods for Restoring the Security Server Database	179
Database Restore Script	179
Database Backup and Restore Utility	179
 Appendix D Cron Expressions	 183
Overview	185
Descriptions and Examples of Cron Expressions	185
 Appendix E Well-Known Attributes	 189
Overview	191
Well-Known Attributes	191
Security Realms and Applications	192
 Appendix F Securing CSM or Back Office Resources (CV)....	 195
Overview	197
Generic Authorization Policy Spreadsheets	197
Trimming Spreadsheet Rows	198
Types of Resources	199

Valid Actions Based on Resource Type	200
Finding a Resource to Secure	200
Bulk Loading and Publishing the Policy	201
Verifying the Policy	201
Appendix G Securing Security GUI Resources	203
Overview	205
Default Authorization Policy Spreadsheet	205
Types of Resources and their Alignment with the Security GUI	206
Editing Spreadsheet Rows	208
Bulk Loading and Publishing the Policy	209
Verifying the Policy	209
Index	211

Figures

Figure 1	CLI add_realm Example	19
Figure 2	CLI add_group Example	19
Figure 3	CLI modify_realm Example	20
Figure 4	CLI modify_group Example	21
Figure 5	CLI remove_group Example	21
Figure 6	CLI remove_realm Example	22
Figure 7	CLI add_role Example	23
Figure 8	CLI remove_role Example	24
Figure 9	CLI find_users Example	24
Figure 10	CLI list_users Example	25
Figure 11	CLI list_attributes Example	25
Figure 12	CLI add_user Example	26
Figure 13	CLI lock_user and unlock_user Examples	27
Figure 14	CLI modify_user Example	28
Figure 15	CLI remove_user Example	28
Figure 16	CLI list_purged_users Example	29
Figure 17	CLI disable_user and enable_user Examples	29
Figure 18	CLI modify_group Example to Define a Login Window	30
Figure 19	CLI change_password Example	30
Figure 20	CLI reset_password Example	31
Figure 21	CLI list_auth_rule Example	39
Figure 22	CLI create_auth_rule Example	39
Figure 23	CLI modify_auth_rule Example	40
Figure 24	CLI remove_auth_rule Example	40
Figure 25	CLI list_auth_policy Example	41
Figure 26	CLI create_auth_policy Example	42
Figure 27	CLI modify_auth_policy Example	43
Figure 28	CLI remove_auth_policy Example	43
Figure 29	CLI publish_policy Example	44
Figure 30	CLI resync_policy Example	45
Figure 31	CLI build_report Example	50
Figure 32	CLI list_keys Example	65
Figure 33	CLI create_key Example	65
Figure 34	CLI disable_key Example	66
Figure 35	CLI enable_key Example	66
Figure 36	CLI delete_key Example	66
Figure 37	CLI list_credential Example	67
Figure 38	CLI store_credential Database Example	67
Figure 39	CLI store_credential Network Example	68
Figure 40	CLI remove_credential Database Example	68
Figure 41	CLI remove_credential Network Example	69
Figure 42	CLI publish_credential Example	70
Figure 43	Security GUI — Introductory Welcome Page	75
Figure 44	Security GUI — Login Page	76
Figure 45	Security GUI — Home Page	76
Figure 46	Security GUI — Identity Management Page	77
Figure 47	Security GUI — Users Tab Showing Users for a Realm	78
Figure 48	Security GUI — Edit User Details Tab (Modifying a User)	79
Figure 49	Security GUI — User Attributes Tab, Current Attributes (Modifying a User)	79
Figure 50	Security GUI — User Attributes Tab, Add Attributes (Modifying a User)	80
Figure 51	Security GUI — Users Tab (Creating a User)	81

Figure 52	Security GUI — Add User Details Tab (Creating a User)	81
Figure 53	Security GUI — User Attributes Tab, New Attributes (Creating a User)	83
Figure 54	Security GUI — User Attributes Tab, Current Attributes (Creating a User)	83
Figure 55	Security GUI — Fields to Search for Users across Realms	84
Figure 56	Security GUI — Groups Tab Showing Groups for a Realm	85
Figure 57	Security GUI — Edit Group Details Tab (Modifying a Group)	85
Figure 58	Security GUI — Group Attributes Tab, Current Attributes (Modifying a Group) ..	86
Figure 59	Security GUI — Group Attributes Tab, Add Attributes (Modifying a Group)	87
Figure 60	Security GUI — Groups Tab (Creating a Group)	88
Figure 61	Security GUI — Add Group Details Tab (Creating a Group)	88
Figure 62	Security GUI — Group Attributes Tab, Add Attributes (Creating a Group)	89
Figure 63	Security GUI — Group Attributes Tab, Current Attributes (Creating a Group)	90
Figure 64	Security GUI — Roles Tab	91
Figure 65	Security GUI — Realms Tab	92
Figure 66	Security GUI — Edit Realm Tab (Modifying a Realm)	93
Figure 67	Security GUI — Realm Attributes Tab, Current Attributes (Modifying a Realm) ..	93
Figure 68	Security GUI — Realm Attributes Tab, Add Attributes (Modifying a Realm)	94
Figure 69	Security GUI — Password Policy Tab (Modifying a Realm)	94
Figure 70	Security GUI — Realms Tab (Creating a Realm)	96
Figure 71	Security GUI — Add Realm Tab (Creating a Realm)	96
Figure 72	Security GUI — Realm Attributes Tab, New Attributes (Creating a Realm)	97
Figure 73	Security GUI — Realm Attributes Tab, Current Attributes (Creating a Realm) ...	97
Figure 74	Security GUI — Password Policy Tab (Creating a Realm)	98
Figure 75	Security GUI — Keys Tab	99
Figure 76	Security GUI — Add Key Details Tab	100
Figure 77	Security GUI — Policy Tab	101
Figure 78	Security GUI — Policy Tab Showing Policies for a Realm	102
Figure 79	Security GUI — Edit Policy Details Tab (Modifying a Policy)	102
Figure 80	Security GUI — Policy Tab (Creating a Policy)	104
Figure 81	Security GUI — Add Policy Details Tab (Creating a Policy)	104
Figure 82	Security GUI — Publish Policy Details Tab	106
Figure 83	Security GUI — Publish Policy Details Tab (Publish/Resynchronize Results) ..	106
Figure 84	Security GUI — Rules Tab Showing Current Rules	107
Figure 85	Security GUI — Edit Rule Tab (Modifying a Rule)	107
Figure 86	Security GUI — Rules Tab (Creating a Rule)	109
Figure 87	Security GUI — Add Rule Tab (Creating a Rule)	109
Figure 88	Security GUI — Audit Records Tab	110
Figure 89	Security GUI — Credentials, Database Tab	111
Figure 90	Security GUI — Add Credential Details Tab (Creating Database Credentials) .	112
Figure 91	Security GUI — Publish Credential Details Tab	113
Figure 92	Security GUI — Credentials, Network Tab	114
Figure 93	Security GUI — Add Network Credential Tab	115
Figure 94	Database Backup and Recovery Utility — Initial Screen	180
Figure 95	Database Backup and Recovery Utility — Main Menu	180
Figure 96	Database Backup and Recovery Utility — Database Restore Menu	181
Figure 97	Database Backup and Recovery Utility — Recovery Options Menu	181
Figure 98	Database Backup and Recovery Utility — Recovery Confirmation	182
Figure 99	Generic CSM Policy Spreadsheet — Initial Rows	197
Figure 100	Generic CSM Policy Spreadsheet — More Rows	198
Figure 101	SERVICE_RSRC and ATTRIBUTE_RSRC Resources	199
Figure 102	ACTION_RSRC Resources	200
Figure 103	CSM Log File	200
Figure 104	Default Security GUI Policy Spreadsheet — Representative Rows	206
Figure 105	Security GUI — All Permissions for the ^Identity.users\$ Resource	207
Figure 106	Security GUI — Only View Permission for the ^Identity.users\$ Resource	207

Tables

Table 1	Notational Conventions.....	xv
Table 2	Labels in Markers	xvi
Table 3	Types of Markers	xvi
Table 4	Default Password Policy	18
Table 5	Add User Details Tab — Field Descriptions	82
Table 6	Add Group Details Tab — Field Descriptions	88
Table 7	Password Policy Tab — Field Descriptions	98
Table 8	Add Key Details Tab — Field Descriptions	100
Table 9	Policy Details Tab — Field Descriptions	104
Table 10	Add Rule Tab — Field Descriptions.....	109
Table 11	Add Credential Details Tab — Field Descriptions.....	112
Table 12	Publish Credential Details Tab — Field Descriptions	113
Table 13	Add Network Credential Tab — Field Descriptions	115
Table 14	Database Field Types.....	142
Table 15	Fields in Cron Expressions	185
Table 16	Special Characters Used in Cron Expressions.....	185
Table 17	Cron Expression Examples	186
Table 18	Well-Known Attributes	191
Table 19	Security Realms and Applications	192
Table 20	Valid Actions by Resource Type.....	200
Table 21	Resources for Perspectives (Top-Level Tabs) in the Security GUI	207

Notational Conventions



Useful information appears in this format.



Provides direction to important information



Important information appears in this format.



Indicates possible risk of damage to data, software, or hardware.



Indicates serious risk of damage to data, software, or hardware.

Table 1 Notational Conventions

Notation	Explanation of Convention
<i>References to printed documents</i>	<i>Helvetica italic</i> Example: See <i>Database Reference Volume 2</i> .
<KEYS>	UPPERCASE HELVETICA, in angle brackets Example: Press <CTRL><Q><SHIFT><P> to create an em dash.
User-entered text	Courier bold Example: Enter Total Charges in the field.
<i>Placeholders for user-determined text</i>	<i>Courier italic</i> , in angle brackets Example: Enter your <password>.
Code samples, TABLE_NAMES, field_names, file and directory names, file contents, user names, passwords, UNIX ENVIRONMENT_VARIABLES	Courier
<i>Placeholders for system-generated text</i>	<i>Helvetica italic</i> Example: Messages appear in this form: <i>timestamp messageId >> text</i> .
Buttons, Icon Names, and Menu items	Helvetica bold Example: Choose Reports from the main menu.

Special Markers

The Comverse ONE Billing and Active Customer Management solution has the three derivatives shown in [Table 2, “Labels in Markers.”](#) For user convenience, any content that is specifically included in a derivative is highlighted with special markers so that it can readily be distinguished.

Table 2 Labels in Markers

Derivative	Label Shown in Markers
Comverse ONE Converged Billing derivative	Converged only
Comverse ONE Real-Time Charging derivative	Real Time only
Comverse ONE Postpaid Billing derivative	Postpaid only

Each derivative has a set of three color-coded markers, as shown in [Table 3, “Types of Markers.”](#) The markers are used individually or in combination to highlight derivative-specific content by:

- Entire chapters
- Selected portions of chapters
- Tables, either entire or partial

Table 3 Types of Markers

Marker	Example	Description
Alert		<ul style="list-style-type: none"> ■ Placed at the beginning of an entire chapter that pertains only to a specific derivative. ■ Placed just before a table that partially or entirely pertains only to a specific derivative.
Block		A shaded box that encloses sections of documentation that pertain only to a specific derivative.
Flag		<ul style="list-style-type: none"> ■ Designates a shaded table row whose contents pertain only to a specific derivative. ■ In a bulleted list, designates an item that pertains only to a specific derivative.

Comverse ONE Documentation List



NOTE

this is a comprehensive list. As such, it may include documentation for products which you have not licensed.

The documents described below reference the Comverse ONE solution products. All documentation available with the Comverse ONE solution is described in the following pages, organized by the following categories:

- Infrastructure Domain
- Rating, Charging, and Promotions Domain
- Billing and Financials Domain (Converged only)
- Customer and Order Management Domain (Converged only)
 - Customer Relationship Management
(Sales Force Automation, Case Management, Campaign Management)
- Mediation and Roaming Solutions Domain
- Self-Service Solutions Domain



NOTE

Read the relevant Solution Description first to get an overview of your Comverse ONE solution. It gives an overview of the functionality in each product domain and also includes cross-references to the user documentation that provides more detailed information about the functionality.

There are two such documents and they are listed under the Infrastructure Domain heading below.

- *Converged Billing & Active Customer Management Solution Description*
- *Real-Time Billing & Active Customer Management Solution Description*

Infrastructure Domain

Download every document in the Infrastructure domain if you purchase the Comverse ONE solution. Documentation for this domain includes the following (in alphabetical order):

- *Alarms Reference*
Contains tables of alarm IDs, descriptions, likely causes, and recommended resolutions for systems and components.

- ***Back Office Administration GUI Guide***
Provides information about the BackOffice subsystems for Inventory Administration, Address Management and Bulk Operations.
- ***Converged Billing & Active Customer Management Solution Description***
General overview of the Comverse ONE Converged Offer and the functionality available in each domain.
- ***Database Reference***
Describes all database tables and fields in detail.
- ***Disaster Recovery Operations Guide (Optional Module)***
The Disaster Recovery Operations Guide serves as both a technical overview of the optional Disaster Recovery solution and as a guide which details the operational procedures for failover, switchover and switchback provided by the solution.
- ***Glossary***
Provides a list of terms used specifically for the Comverse ONE solution
- ***Investigation Units and Financial GUIs Guide***
Describes the GUI-based tools used for investigating and troubleshooting various financials related processes: payments, bill invoices, refunds, and incomplete data work entries
- ***Operation Reference***
Describes the processes in the Comverse ONE solution.
- ***Platform Operations Guide***
Describes the back-end operations and maintenance functionality of the core Comverse ONE solution components. Includes AIX/HACMP platform and cluster operations, Linux/Veritas platform and cluster operations, backup/recovery, shared storage and fiber switch operations, and tape backup operations.
- ***Product Catalog Overview***
Provides a high-level description of the Comverse ONE solution Product Catalog, which is the primary mechanism for creating, configuring, managing, and propagating Product Catalog versions.
- ***Product Catalog User Guide***
Instructions on using the Product Catalog application to define and manage all aspects of Service provisioning.
- ***Real-Time Billing & Active Customer Management Description***
General overview of the Comverse ONE Real-Time Offer and the functionality available in each domain.
- ***Schedulable Entity Reference Manual***
Documents all the jobs, monitors, and workflows, for each component.
- ***Security Platform Operations Guide***
Technical overview of the security platform and information on how to provision and administer the platform.
- ***Security Server API Guide***
Provides an overview of the interfaces exposed by the Java-based Security SDK API, which client applications can leverage to access various security services, such as authentication, authorization, auditing, key management, and credentials management. Also provides information on the Security Web Services API, which provides interfaces to a subset of Security Server commands (Identity Management commands).
- ***Signaling Gateway Unit Guide***
Describes the hardware, installation, configuration, and maintenance of the Signaling Gateway Unit (SGU) used to connect Comverse real-time systems to the SS7 signaling network using either traditional SS7 protocols or Sigtran (SS7 over IP).
- ***System Measurements Guide***
The Comverse ONE Solution automatically collects statistical data from the Service Logic

Unit (SLU) and the Service Gateway Unit (SGU). This includes service statistics on the SLF layer and platform data on the IPF layer.

This guide describes the format and location of this measurement information and provides a description of the meaning of the data. The measurement data can be used to create reports. It can also be imported into other applications (such as Excel) to be viewed.

- ***Unified API Guide***

General overview of the Unified API, a brief description of its architecture, and information about:

- Framework classes and the functionality they provide
- Two standard interfaces provided with the Unified API (client SDK and web services)
- A subset of Unified API business methods most commonly used

- ***Unified Platform Guide***

Technical overview of the Unified Platform and information on the procedures to manage core systems operations in the Comverse ONE solution.

Rating, Charging, and Promotions Domain

Documentation for this domain includes the following (in alphabetical order):

- ***Bulk Provisioning Guide***

- The *CC Batch* utility enables bulk creation of recharge vouchers and subscribers.
- The *Bulk Provisioning* Utility enables bulk creation of anonymous accounts to support the pre-activation of pre-paid SIM cards.

- ***Call Flows Reference***

Call flows detail the logic flow of specific scenarios. Multiple access numbers can map to the same call flow. Different resellers have the option to publish different numbers but share the same logic.

- ***Charging Interfaces Guide***

Describes the four interfaces that enable external services to support real-time authorization, rating, and charging for transactional usage: (1) the Event Charging Interface, a simple TCP/IP-based interface, (2) Open Services Access (OSA), (3) a Diameter-based interface version enhanced to take advantage of features of the Comverse ONE solution, and (4) a Diameter-based interface packet-switched version.

- ***Customer Care Client Provisioning Guide — Real-Time***

Detailed task-oriented instructions for using Customer Care Client.

- ***Diameter Gateway Unit Guide***

Describes the hardware, installation, configuration and maintenance of the Diameter Gateway Unit (DGU) used to connect Comverse real-time systems to external services, using the diameter protocol over IP.

- ***Network Interfaces and Notifications Guide***

Describes the operation, features, and provisioning of notifications, CAMEL-enabled services, and USSD-enabled services.

- ***Network Self-Care Guide***

Describes the configuration, structure, and features.

- ***Rating Technical Reference***

Describes the Unified Rating Engine, which is the subsystem responsible for gathering incoming CDRs and processing them for billing.

- ***Reports and Data Extracts Guide — Real-Time***

Describes the real-time Operational Reports Interface (ORI) and the Data Warehouse Extract Utility.

- ***Recurring–Non-Recurring Charges Server Guide***
Describes all processes commonly available through the Recurring —Non-Recurring Charges Server.
- ***Voucher and Recharge Guide***
Describes the process by which subscribers add funds to accounts using recharge vouchers through IVR, interaction with Customer Service, and other methods. Provides details of the Recharge Control Table, which allows resellers to provision the effects of recharges so that bonuses, discounts, and other changes to offers can result from a successful recharge. Also describes the Card Generator software used to create batches of recharge vouchers.

Billing and Financials Domain (Converged only)

Documentation for this domain includes the following (in alphabetical order):

- ***Advanced Statement Numbering Guide***
Describes how to configure and use Advanced Statement Numbering.
- ***Billing Reports and File Layouts User Guide***
Describes control reports and other file formats.
- ***Billing Technical Reference***
High-level descriptions of billing architecture, administration, bill generation and formatting, and system parameters
- ***Collections Guide***
Contains information on configuring Collections database tables, running the Collections module, and using the Collections interface.
- ***Invoice Designer Strings and Filters Reference***
Describes the static strings, dynamic strings, and filters in the Invoice Designer.
- ***Invoice Designer Technical Reference***
Describes how to configure and run Invoice Designer.
- ***Invoice Designer User Guide***
Describes the Invoice Designer and how to perform the tasks needed to create an invoice template.
- ***Journals Guide***
Describes the theory, configuration, and running of Journals processes.
- ***Miscellaneous Configurable Entities***
Instructions for configuring late fees, adjustments, and several other database entities used in postpaid and converged billing.
- ***Process Workflow Orchestration Guide***
Describes the command-line entries and the default queries for running billing-related processes via the Unified Platform.
- ***Taxation Guide***
Describes the configuration, operation, structure, and features of Taxation.

Customer and Order Management Domain (Converged only)

Documentation for this domain includes the following (in alphabetical order):

- *Application Integrator Adapter Developer Kit User Guide*
Provides information necessary for the development of custom Application Integrator adapters.
- *Application Integrator Add/Copy Header User Guide*
Describes the adapter that adds or copies header information in messages.
- *Application Integrator Aggregator Adapter User Guide*
Describes the adapter that aggregates multiple input messages as a single composite output message.
- *Application Integrator File Adapter User Guide*
Describes the configuration process and rules for the file adapter.
- *Application Integrator CORBA Adapter (JacORB) User Guide*
Describes the elements and uses of the Application Integrator client and server Common Object Request Broker Architecture (CORBA) adapters for JacORB.
- *Application Integrator Filemover Adapter User Guide*
Describes the use and configuration of the adapter, which is used to copy or move files from one machine to another.
- *Application Integrator Generic Services User Guide*
Describes the Null adapter, Trash adapter, and Initiator adapter generic services.
- *Application Integrator HTTP Adapter User Guide*
Describes the use and configuration of the adapter which provides an interface between HTTP clients and the ApplicationIntegrator.
- *Application Integrator IPDR Adapter User Guide*
Describes use and configuration of the I adapter which converts the “compact encoding” form of IPDR billing record documents into a form easily parsed by the ApplicationIntegrator message broker.
- *Application Integrator JMS Adapter User Guide*
Describes the use and configuration of the adapter, which is used with edge systems that transmit or receive JMS messages.
- *Application Integrator KSI Adapter User Guide*
Describes the use and configuration of the adapter, which is used with edge systems that transmit or receive data formatted according to the Kenan Standard Interface (KSI) protocol.
- *Application Integrator Operator Guide*
Describes the commands that operate the Application Integrator at creation and runtime.
- *Application Integrator Python Adapter User Guide*
Describes the use and configuration of the adapter, which enables a user to run a *Python* script from within an integration.
- *Application Integrator Retry Adapter User Guide*
Describes the use of the a dapter to resend messages in case of failed transmissions.
- *Application Integrator SAS Adapter User Guide*
Describes the use and configuration of the adapter, which is used with edge systems that transmit or receive data formatted according to the *Comptel*/Mediation Device Solutions/Subscriber Administration System (MDS/SAS) protocol.
- *Application Integrator Sequence Adapter User Guide*
Describes the use of the adapter to generate unique sequence numbers for messages.
- *Application Integrator System Administrator Guide*
Outlines installation, sizing, operation, and administration of the Application Integrator

and logging. Describes configuration of the user environment and commands for creation and operation of the Application Integrator.

- *Application Integrator Unified API Client Adapter User Guide*
Describes the adapter which is used for interfaces based on the Unified API Client.
- *Application Integrator Unified API Server Adapter User Guide*
Describes the adapter which is used for interfaces based on the Unified API Server.
- *Application Integrator URL Client Adapter User Guide*
Describes the use and configuration of the adapter which makes it possible for a client to gain access to many kinds of network-accessible resources that are identified by a URL.
- *Application Integrator User Guide*
Describes creating integration specifications, creating instances of the Application Integrator, and commands for operation of the Application Integrator. Provides a complete user guide for the iMaker compiler.
- *Application Integrator XSLT User Guide*
Describes the use and configuration of the adapter which is used with applications (sometimes called edge systems) that transmit or receive XML-formatted data.
- *Customer Center User Guide*
Detailed task-oriented instructions for using Customer Center.
- *Inventory Guide*
Describes the configuration, operation, structure, and features of Inventory.
- *Inventory Replenishment Guide*
Describes the operation, structure, and features of Inventory Replenishment.
- *Orders Services Guide*
Describes the structure and features of Orders Services.
- *Request Handling and Tracking and Service Fulfillment User Guide*
Describes the configuration, operation, structure and features of Request Handling and Tracking and Service Fulfillment.
- *Workflow Developers Guide*
Helps new users understand the rules-based business process management system so users can create solutions and integrate Workpoint within those solutions.
- *Workflow User Guide*
Describes the configuration, operation, structure, and features of Workpoint.

Customer Relationship Management

- ***Billing Reports and File Layouts User Guide***
Describes control reports and other file formats.
- ***Campaign Management Data Mapping Reference***
Describes how the data in DataMart is mapped to information in the Comverse ONE Customer database, the Comverse ONE ODS, and the Comverse ONE Sales and Service database.
- ***Campaign Management DataMart Reference***
Contains in-depth technical information on how to configure and populate the data mart used by all Campaign Management applications.
- ***Campaign Management Outbound Marketing Manager Reference***
Describes how to use the Campaign Management Outbound Marketing Manager features and guides you through the program's basic functionality.
- ***Campaign Management Quick Implementation Guide***
Helps novice users get started with implementing Campaign Management. It contains an overview of the product architecture, information on data mart design and creation, an explanation of how extraction works, and procedures for creating web pages, reports, lists, and campaigns.
- ***Campaign Management Topic Implementation Guide***
Provides information for implementers and professional services personnel who are creating applications that will run on an Campaign Management EpiCenter. Summarizes the Campaign Management functionality, architecture, and administration and contains in-depth technical information for configuring the Campaign Management topics required for Campaign Management and analysis.
- ***Campaign Management User Guide***
Provides you with basic information about the Campaign Management applications.
- ***Case Management User and Administration Guide***
Contains detailed information about GUI screens and form fields that appear in the Case Management application. Also provides information on performing general procedures in the GUI and administrative tasks.
- ***Customer Center User Guide***
Detailed task-oriented instructions for using Customer Center.
- ***Sales and Service Admin Console User Guide***
Provides supervisors, managers, and executives with the information to use the Case Management and Sales Force Automation Admin Console application.
- ***Sales and Service Application Reference*** Contains technical reference information relevant to implementers involved in implementing and customizing CRM applications at customer sites. This book provides the reference context for the procedural information available in the Implementation Guide.
- ***Sales and Service Architecture Reference***
Provides technical information relevant to individuals involved in implementing the Open Architecture and the applications built on the architecture
- ***Sales and Service Data Dictionary Reference***
Includes a listing and description of the tables and columns used to store CRM operational business data. It also includes a description of the naming conventions for the tables. The target audience includes database administrators, application developers, and implementers.
- ***Sales and Service IBR Designer User Guide***
Describes how to use the IBR Designer to create Intelligent Business Rules, which can be used to implement rule-based behavior within your CRM applications.

- ***Sales and Service Implementation Guide***
Provides procedural information relevant to individuals involved in implementing and customizing the core and the Sales and Service applications built on the core.
- ***Sales and Service Integration Guide***
Provides overview and configuration information for the set of tools used to exchange data with a variety of back-end data sources, including generic SQL sources, Java and EJB-based sources, Web services, and other database types.
- ***Sales and Service Workflow Designer***
Explains how to use Workflow Designer, a web-based graphical tool for defining and editing workflows
- ***Sales Force Automation User and Administration Guide***
Contains detailed information about GUI screens and form fields that appear in the Sales Force Automation application. Also provides information on performing general procedures in the GUI and administrative tasks.

Mediation and Roaming Solutions Domain

Documentation for this domain is subdivided into Mediation/Roaming and Revenue Settlements.

Mediation and Roaming

Mediation and Roaming documentation includes the following (in alphabetical order):

- ***Collection API Guide***
Provides the concepts and functions for the Collection Application Programming Interface (CAPI).
- ***Data Manager GUI Reference***
Contains detailed information about GUI screens and form fields that appear in the Data Manager interface
- ***GRID Mapping Language Developer Guide***
Describes the mediation feature components, semantics, and general syntax of the GRID Mapping Language (GML).
- ***Installation Guide for HP***
Describes how to install and configure the application, components, and some third-party applications associated with the HP platform.
- ***Installation Guide for HP Itanium***
Describes how to install and configure the application, components, and some third-party applications associated with the HP Itanium platform.
- ***Installation Guide for HP PA-RISC***
Describes how to install and configure the application, components, and some third-party applications associated with the HP PA-RISC platform.
- ***Installation Guide for IBM***
Describes how to install and configure the application, components, and some third-party applications associated with the IBM platform.
- ***Installation Guide for SUN***
Describes how to install and configure the application, components, and some third-party applications associated with the SUN platform.
- ***Mediation and Roaming User Guide***
Provides information on how to use the GUI interface, including information on using the Data System Manager application pages.
- ***Mediation API Guide***
Contains reference information on using the Mediation API.

- ***Roaming Database Reference***
Provides reference information on the Roaming database.
- ***Roaming Setup Guide***
Describes how to configure the Roaming Setup application pages. It also provides information on working with TAP, RAP, and CIBER statistics.
- ***Scripts Guide***
Provides information on script files, which contain additional instructions on functions for data collection and transmission.
- ***Socket-Based API Guide***
Explains the building applications using the Socket-Based Record Transmission API. Programmers can use the guide to use the records received from the Data system for their own customized downstream application solutions.
- ***System Manager GUI Reference***
Contains detailed information about GUI screens and form fields that appear in the System Manager interface
- ***Variable-Length GRID Guide***
Provides information on how to configure the control files for variable-length GRID.

Revenue Settlements

Revenue Settlements documentation includes the following (in alphabetical order):

- ***Comverse Revenue Settlements Billing System Adapter Guide***
Describes the configuration, operation, and installation for the Billing System adapter.
- ***Comverse Revenue Settlements Data Model Guide***
Overview of data model entities (such as partners, accounts, revenue sharing, and rate schedules) and how to configure them in the database.
- ***Comverse Revenue Settlements Database Reference***
Detailed descriptions of fields and tables in the database.
- ***Comverse Revenue Settlements Technical Reference***
Instructions for installing and operating Revenue Settlements. Also contains processing descriptions.
- ***Comverse Revenue Settlements User Guide***
Instructions for using the Revenue Settlements GUI.

Self-Service Solutions Domain

The Comverse ONE Self-Service Solutions domain consists of the core products plus the optional separately licensed premium products. The core products consist of the following:

- Self-Service Solutions Platform
- Self-Service Solutions Applications

Self-Service Solutions Platform Documentation

The Self-Service Solutions Platform has a comprehensive set of documentation covering the installation, configuration, and use of our products. The documentation set is divided into the following categories:

- **Manuals:** These manuals cover installing and using the platform.
- **Reference:** These reference documents contain information about APIs, databases, configuration files, and so on. These documents are delivered in HTML.

Self-Service Solutions Platform Manuals

Self-Service Solutions Platform manuals include the following (in alphabetical order):

- ***Administration Guide***
Provides operations and maintenance instructions for Web applications using the Self-Service Solutions Platform.
- ***Communications Billing and Usage Reference***
Provides detailed descriptions of the data models and structure of the Self-Service Solutions Platform Communications Billing and Usage (CBU) database.
- ***Connectors Development Guide***
Provides instructions for developing and customizing Connectors of the Self-Service Solutions Platform.
- ***Core Module Development Guide***
Provides instructions for configuring and developing features of the core module of the Self-Service Solutions Platform.
- ***Customer Interaction Datastore Reference***
Provides detailed descriptions of the data models and the structure of the Self-Service Solutions Platform Customer Interaction Datastore (CID).
- ***Database Modules Development Guide***
Provides instructions for configuring, customizing, and developing features of the database module of the Self-Service Solutions Platform.
- ***Platform Installation Guide***
Provides installation and configuration instructions for the Self-Service Solutions Platform.
- ***Platform Services Guide***
Provides instructions for configuring, customizing, and developing features that use the services provided by the Self-Service Solutions Platform.
- ***Processors Development Guide***
Provides instructions for developing and customizing Processors of the Self-Service Solutions Platform.
- ***Reports Development Guide***
Provides instructions for developing and customizing Reports of the Self-Service Solutions Platform.
- ***Self-Service Solutions Overview Guide***
Provides a high-level architectural and functional description of the Comverse ONE Self-Service Solutions. It also includes a detailed description of the concepts and development process to create and deploy Self-Service Solutions.
- ***Web Applications Development Guide***
Provides instructions for configuring, developing, and deploying Web applications that use the Self-Service Solutions Platform.

Self-Service Solutions Platform Reference

Self-Service Solutions Platform reference documentation includes the following (in alphabetical order):

- ***Base Logic Manager Reference***
Describes usage syntax and configuration files for the Base Logic Manager (BLM) APIs. These APIs are the core services of the Self-Service Solutions Platform.
- ***CID2CBU Object Mapping Reference***
Describes the default mapping of Customer Interaction Datastore (CID) and Communications Billing and Usage (CBU) objects.

- **Communications Billing and Usage Reference**
Provides detailed descriptions of fields and tables in the Communications Billing and Usage (CBU) database.
 - **Customer Interaction Datastore Reference**
Provides detailed descriptions of fields and tables in the Customer Interaction Datastore (CID).
 - **Integration Services Framework API Reference**
Describes usage syntax of the set of APIs to program connectors and other components of the Intelligent Synchronization Framework (ISF).
 - **Integration Services Framework Message Cache Reference**
Provides detailed descriptions of fields and tables in the Intelligent Synchronization Framework (ISF) Message Cache.
 - **Integration Services Framework Script API Reference**
Describes usage syntax of the Intelligent Synchronization Framework (ISF) script APIs to program the ISF connectors.
 - **JavaServer Page Framework for Internet Application API Reference**
Describes usage syntax for the JavaServer Page Framework for Internet Application (JFN) APIs. These APIs are used to build JSPs using the JFN. This framework provides basic application functions and services as the foundation of user interfaces.
 - **Logger Message Reference**
Provides detailed descriptions of the Self-Service Solutions Platform log messages.
 - **QRA API Reference**
Describes usage syntax for the Query, Reporting, and Analysis (QRA) Engine APIs. These APIs are used to build reports.
 - **UTIL API Reference**
Describes usage syntax for the UTIL package used by different components of the Self-Service Solutions Platform. This package contains a set of utilities including the logger.
- Self-Service Solutions Applications Documentation

Each Self-Service Solutions Application comes with a comprehensive set of documentation covering the installation, configuration, and use of the product. The application documentation expands and complements the Self-Service Solutions Platform documentation.

The documentation set is divided into the following categories:

- **Manuals:** These manuals cover installing and using the application.
- **Reference:** These reference documents contain information about APIs, databases, configuration files, and so on. These documents are delivered in HTML.

Self-Service Solutions Application Manuals

A full set of these manuals is available for each Self-Service Solutions Application. The documentation set includes the following (in alphabetical order):

- **Business Objects Model Reference**
Provides a detailed description of the models and entities that make up the Self-Service Solutions Application.
- **Catalog Loader Reference**
Provides information about the Catalog Loader, including a functional description as well as installation, configuration, and use instructions.
- **Configuration and Development Guide**
Provides instructions for configuring and developing Self-Service Solutions Application features.
- **Feature Reference**
Describes the logic and provides use cases for the functional domains of the application.

- ***Out-of-the-Box Reference Guide***
Describes the Self-Service Solutions Application Out-of-the-Box release.
- ***Self-Service Installation Guide for Comverse ONE***
Provides detailed installation, configuration, and deployment instructions for the Self-Service Solutions Application alongside other elements of the Comverse ONE solution.
- ***Self-Service Installation and Deployment Guide***
Provides detailed installation, configuration, and deployment instructions for the Self-Service Solutions Application.
- ***Introduction***
Provides a high-level architectural and functional description of the Self-Service Solutions Application. It covers common features, order management, account management, and bill presentment.

Self-Service Solutions Application References

A full set of these references is available for each Self-Service Solutions Application. The reference documentation set includes the following (in alphabetical order):

- ***API Reference***
Describes usage syntax for the Self-Service Solutions Application APIs. These APIs are used to program the user interface and manage data.
- ***Invoice Schema Reference***
Describes the invoice schema reference of the Self-Service Solutions Application.
- ***Presentation Layer Page Flow Reference***
Describes the page flows of the Self-Service Solutions Application.
- ***Specification Entity Relationship Diagrams***
Provides diagrams describing the actors, use cases, user activity, and storyboard in IBM Rational Rose format.

Self-Service Solutions - Separately Licensed Products

Documentation available with optional, separately-licensed premium products in the Comverse Self-Service Solutions is listed below.

Online Catalog Manager

Online Catalog Manager (OCM) documentation includes the following (in alphabetical order):

- ***Introduction to the Online Catalog Manager***
Provides a high-level architectural and functional description of the Online Catalog Manager.
- ***Online Catalog Manager Getting Started Guide***
Describes the best way to build product catalogs in the Online Catalog Manager. This manual is a template for creating end-user documentation.
- ***Online Catalog Manager Installation and Configuration Guide***
Provides installation and configuration instructions for the Online Catalog Manager.
- ***Online Catalog Manager User Documentation Template***
Describes the use of the Online Catalog Manager. This manual is a template for creating end-user documentation. This manual covers many common concepts and procedures of the OCM.
- ***Online Catalog Manager User Guide***
Provides a detailed description of the concepts and use of the Online Catalog Manager. The topics include:
 - Managing Media Files

- Managing Offers
- Managing Prices
- Managing Products
- Managing Properties
- Managing Reference Data
- Publishing

CSR Portal

The CSR Portal product includes the standard Application documentation, plus the following manual:

- *CSR Portal User Guide*
A guide to using the CSR Portal UI.

Chapter 1

Introduction

1

orded (sender
The desti
notifying
ng The noti
ieve The m
t access To

e

Welcome

Welcome to the *Security Platform Operations Guide* for the Comverse ONE Billing and Active Customer Management solution.

The purpose of this manual is to convey Comverse's solution to address our client organizations' Sarbanes-Oxley Act of 2002 (SARBOX) compliance requirements and the overall security enhancements for the Comverse ONE solution. The move to enhance the general security of Comverse ONE solution components resulted from the ongoing trend for client organizations to become or remain SARBOX compliant. In response to this need, Comverse undertook development of a security infrastructure to support our clients' efforts.

The Comverse security platform is Comverse's solution to address the SARBOX compliance requirements of our client organizations and to enhance the general security aspect of the Comverse ONE solution. The platform provides authentication, authorization, and accounting (AAA) capabilities that allow client organizations to centrally manage Comverse ONE application user details, centrally manage application access control policies, and obtain a central view of audit information that can be used to comply with certain SARBOX regulatory requirements. The platform is built using an open framework and uses industry-standard interfaces for AAA services. The security platform can also provide these services for other components in a client organization's infrastructure, enabling the Comverse security platform to provide a central security solution for client organizations.

The *Security Platform Operations Guide* serves as both a technical overview of the security platform and a guide on how to provision and administer the platform.

New Features for This Release

The following is a list of new features in the Comverse ONE 3.5.50 release that impact this document:

- Ability to secure resources for the Web-based Security GUI

Who Should Use This Document

This document is intended for use by security administrators who are responsible for all aspects of provisioning and administering the security platform.

To get the most from this document, it is helpful to become familiar with the overall security solution and how it functions. See [Chapter 2, "Security Overview."](#) It is also beneficial to have a good familiarity with the Unified Platform. See the *Unified Platform Guide* for information.

Organization of This Document

The remainder of this document is organized as follows:

[Chapter 2, "Security Overview,"](#) provides a technical overview of the security solution.

[Chapter 3, "Identity Management,"](#) provides background information about identity management and provides configuration instructions for security realms, realm groups, roles, and user accounts.

[Chapter 4, "Policy Management,"](#) provides background information about policy management and provides configuration instructions for authorization policies and rules.

[Chapter 5, "Audit Management,"](#) provides background information about audit management, provides instructions for audit management operations, describes the audit record format, and provides details on events and event outcome codes used in audit records.

[Chapter 6, “Encryption Key and Credentials Management.”](#) provides background information about encryption key and credentials (database passwords and SNMP community strings) management and provides instructions for symmetric key and credentials management operations.

[Chapter 7, “Security GUI.”](#) provides information on using the Web-based Security Server graphical user interface (GUI).

[Chapter 8, “Database Reference.”](#) provides detailed information about the Security Server database tables.

[Appendix A, “Security-Related Management Shell Commands.”](#) provides a summary of all security-related commands executed using the Management Shell command line interface (CLI), including all command-specific options.

[Appendix B, “Attribute Conflict-Resolution Rules.”](#) provides an explanation of the rules used to resolve conflicts among custom attributes defined at the security realm, realm group, and user levels.

[Appendix C, “Security Server Database Restore Operations.”](#) provides (1) summary information about the automated data export/database backup for the Security Server database and (2) more detailed information about the manual database restore operations.

[Appendix D, “Cron Expressions.”](#) provides general information about `cron` expressions, which are used to establish login windows.

[Appendix E, “Well-Known Attributes.”](#) provides information about the well-known attributes (system-defined attributes) in the Comverse ONE system.

[Appendix F, “Securing CSM or Back Office Resources \(CV\).”](#) provides information on how to secure access to functionality/fields in the Customer Center GUI and the Back Office GUI.

[Appendix G, “Securing Security GUI Resources.”](#) provides information on how to secure access to functionality in the Web-based Security GUI.

Chapter 2

Security Overview

2

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e
v

Security Solution Overview

The Comverse ONE solution provides a security platform that supports your organization's efforts to become or remain compliant with the Sarbanes-Oxley Act of 2002 (SARBOX).

The SARBOX legislation calls for public companies to employ sufficient checks and balances for any changes in the system that play a direct or indirect part in financial controls or financial data, and to ensure accurate financial reporting/accounting of financial results from the company. Public companies must ensure that appropriate controls are in place when making any changes to any software that plays a part in financial controls or reports and must ensure that all types of changes are monitored and audited.

The SARBOX legislation is primarily concerned with software that (1) is the source of data used in financial reports, (2) helps manage one of those resources, and (3) is related to finance reporting. The legislation mandates that a check-and-balance system be in place to ensure that any change made to an application does not adversely affect financial controls or reports.

In addition, the SARBOX legislation also requires public companies to audit their SARBOX compliance with an external auditing organization, validate the financial reporting processes (as opposed to the actual financial data), audit internal development controls and processes, and audit any changes to any software system that plays a part in financial controls or reports.

Industry best practices include, but are not limited to, the following: (1) a well-defined process to control development and testing, and to monitor and audit all user activities in the system; (2) a change-control board to control any software system changes, configurations, and patches, and to formally approve those changes with electronic signoffs and authentication; and (3) a method to automate, standardize, and centralize the controls in standard business processes.

The purpose of the security platform and SARBOX in the realm of the Comverse ONE solution is to present sufficient controls to allow client organizations to be SARBOX compliant, address widespread security requirements such as the Payment Card Industry Data Security Standard (PCI DSS) urged by our key clients, and provide an ability to customize the core components of the Comverse ONE solution for specific client needs in relation to SARBOX compliance.

As a summary, the Comverse security platform does the following:

- Supplies appropriate controls to ensure that any changes to the system related to financial control or financial data are monitored and audited.
- Follows industry best practices and standards for authentication, authorization, and accounting (AAA).
- Provides centralized key management for symmetric encryption keys and centralized database password management.
- Provides integration points to allow for interfacing to your organization's existing security infrastructure or, if no infrastructure is in place, provides a security infrastructure for components external to the Comverse ONE system.

The following sections present a technical overview that describes the various services provided by the security platform.

Identity Management and Authentication

Identity management is defined as the management of the identity life cycles of entities (that is, subjects) during which (1) the identity is established, (2) the identity is described, and finally (3) the identity is destroyed. Identity management provides mechanisms to establish (that is, create or modify) and administer identity credentials for the purpose of authenticating users of Comverse ONE client applications such as Product Catalog, Self-Service, and so on.

Authentication is the act of establishing or confirming something (or someone) as authentic — that is, of verifying that claims made by or about the thing are true. When authenticating people, this usually means confirming or verifying their identities via a login mechanism.

Identity management and authentication are often considered the same thing although, in reality, they serve different purposes. As defined above, identity management is concerned with creating, modifying, and destroying the identity, whereas authentication involves confirming or verifying that this person has an identity provisioned on a given system.

Identity Management Framework

The identity management service, which is centrally hosted on the Security Server, exposes a Web services interface for authentication of subjects from various applications. Client applications performing authentication operations send/receive Security Assertion Markup Language (SAML) requests/responses via the Web services interface.



NOTE

A client library, provided by the Security Server Application Programming Interface (API) for client applications, hides the complexities of SAML messaging. Any Java or C++ client application that needs to interface with the Security Server should use the provided client library.

In addition to handling authentication requests from Comverse ONE client applications, the identity management service also supports administrative operations by means of the Security Server administrator interfaces (that is, the command line interface and the graphical user interface).

The Security Server identity management repository is implemented as a relational database. The identity management service uses an object-relational mapping (ORM) application programming interface (API) to map the identity object model to the relational model. If your organization has an existing identity repository, you can still leverage the Security Server. Custom identity repository adapters can be provided to interface with existing identity repositories, such as Lightweight Directory Access Protocol (LDAP) repositories.



NOTE

Custom repository adapters are not part of the core solution and are provided as a customization, based on your organization's requirements.

Identity Management API

As part of a Comverse initiative to use standard interfaces so that client organizations can easily integrate with the security infrastructure, SAML was chosen as the API to perform authentication operations. SAML is an XML-based protocol for communicating user authentication, entitlement, and attribute information.

As previously mentioned, the Security Server API library for Java and C++ client applications allows for easy integration with the identity management service. Knowledge of the SAML protocol is not necessary to use our solution.

Policy Management and Authorization

In its purest form, a policy can be described as “a set of considerations designed to guide decisions of courses of action.” In a practical sense, a policy is a set of rules indicating which subjects are permitted to access which resources using which actions under which conditions. With this understanding of what a policy is, policy management can be defined as the creation, modification, and destruction of the artifact (for example, a file or a relational schema) that defines the policy.

Authorization is the process of determining if a subject, once authenticated, is permitted access to a given resource. Authorization and access control are synonymous in this context because both define what is permitted or denied for a given subject.

When policies are defined for Comverse ONE components, they are associated with security roles, and it is these roles that are allowed or denied access to resources. An individual user’s role, therefore, determines the user’s access privileges. This type of policy definition is considered Role-Based Access Control (RBAC), and it is the primary policy model used by Comverse ONE applications.

Policy Management Framework

The policy management service is a distributed service that adheres to the Extensible Access Control Markup Language (XACML) standard established by the Organization for the Advancement of Structured Information Standards (OASIS). The XACML specification defines an architecture that is divided into the following three components:

- **Policy Administration Point (PAP):** This component is responsible for policy creation. In the Comverse ONE solution, the Security Server acts as the Policy Administration Point. Policy administrators interact with the PAP when defining and publishing a policy.
- **Policy Decision Point (PDP):** This component is responsible for policy evaluation and rendering an authorization decision. The Policy Decision Point can be (1) located with the application, (2) integrated with the Unified Platform Agent (UPA), or (3) centrally located on the Security Server. In normal scenarios, the PDP is located with the UPA on the same machine and is a service of the UPA. See the *Unified Platform Guide* for more information about the UPA.
- **Policy Enforcement Point (PEP):** This component is responsible for access control by making decision requests and enforcing authorization decisions. The Uniform API (sometimes referred to as the Single API, or SAPI) is an example of a Policy Enforcement Point.

These components make up the policy management framework and together provide the access control infrastructure that is required by Comverse ONE applications.

Policy Management API

As part of the initiative to use standard interfaces so that client organizations can easily integrate with the security infrastructure, Extensible Access Control Markup Language (XACML) was chosen as the API for defining authorization policies. XACML provides a policy language that allows for defining the access control requirements for application resources. The language and schema support data types, functions, and combining logic that permit complex (or simple) rules to be defined. XACML also includes an access decision language used to represent the runtime request for a resource. When a policy that protects a resource is located, functions compare attributes in the request against attributes contained in the policy rule, ultimately yielding a permit or deny decision.

Audit Management and Accountability

Audit management is defined as management of the user activities that occur throughout the Comverse ONE solution that directly or indirectly affect financial data or controls. Audit management provides Comverse ONE components the ability to create auditable activities.

Audit Management Framework

The audit management service is a distributed service across the Security Server, the Unified Platform Agent (UPA), and the Security Server API. Components that operate on the Security Server are responsible for central management of audit records and provide a centralized interface to securely view the auditing data. Components that operate on the UPA handle the periodic “checkpointing” (that is, forwarding to central storage) of audit records from the managed node to the Security Server. Finally, the Security Server API provides interfaces for applications to create, modify, and publish auditable events. These events are securely transferred locally to the UPA auditing service by the Security Server API and eventually transferred to the Security Server via the UPA.

Audit Management API

As part of the initiative to use standard interfaces so that client organizations can easily integrate with the security infrastructure, the Distributed Audit Service (XDAS) was chosen as the API for audit record formats and definition. XDAS defines a set of generic auditable events that are relevant for most distributed systems such as the Comverse ONE solution, and defines a common portable audit record format to facilitate merging and analysis of audit information from multiple components at the distributed system level.

Data Encryption and Credentials Management

To complete our suite of security services, the solution provides client applications with an API to easily perform data encryption and provides centralized management of encryption keys and security credentials (that is, database passwords and network-device SNMP community strings). Providing a consistent, standardized way for Comverse ONE applications to perform data encryption simplifies and eliminates inconsistencies in security-related operations throughout the Comverse ONE solution. Also, to improve and simplify how business database credentials are managed for Comverse ONE components, all database credentials (that is, database passwords) are centrally managed at the Security Server. In addition to database credentials, network credentials (that is, SNMP community strings) for network devices are stored in the Security Server database.

Data Encryption and Credentials Management Framework

The data encryption framework is a set of interfaces provided by the Security Server API for the purpose of encryption and symmetric/asymmetric encryption key creation. Application symmetric encryption keys are stored centrally at the Security Server.

The credentials management framework consists of (1) a set of interfaces provided by the Security Server API and (2) a credentials management service hosted on the Security Server that provides credentials (database credentials and network credentials) administration and credentials retrieval. During initialization, applications retrieve the database credentials for one or more target databases. Centrally managing these credentials allows your organization the flexibility to change the Comverse ONE database passwords to comply and align with your own security policies. Changes made to database credentials from the Security Server are propagated to the target databases. Network credentials (SNMP community strings) are used by the Unified

Platform Manager for SNMP authorization. In the future, other credentials such as operating system user passwords will be centrally managed. Currently, only database and network credentials are managed.

Data Encryption and Credentials Management APIs

The data encryption API provides standard encryption algorithms, such as AES (Advanced Encryption Standard), Blowfish, and RSA (Rivest, Shamir, Adleman). APIs for digital signatures, message digests, and password-based encryption are also available.

The credentials management API provides mechanisms to retrieve database passwords from the Security Server to a local cache, and to update and retrieve passwords from the cache. There is also an API that provides an interface for fetching the SNMP community string for a specified network device, which is identified by a combination of node class, node name, and node instance.

Chapter 3

Identity Management

3

orded (sender
The desTir
noTiEying
ng The noTi
ieve The m
cT access To

e

v

Identity Management Overview

Identity management is the management of the “identity life cycle” of subjects during which the following things happen:

- Identity is established.
Establishment of an identity occurs when a name or number is connected to a subject. For example, identity is established when a user such as customer service representative logs in and is authenticated.
- Identity is described.
Description of the identity occurs when one or more attributes that are applicable to this particular subject are assigned to the identity.
- Identity is destroyed.
For example, an identity is destroyed when the user logs out or the session expires.

Identity management can be separated into three perspectives:

- Pure identity paradigm, which provides creation, management, and deletion of identities without regard to access or entitlements.
- User access (that is, authentication) paradigm.
- Service paradigm, where a system provides services to users.

Of the three identity management perspectives, the Comverse Security Server provides the pure identity management and user access capabilities. This chapter describes how to use the Security Server’s identity management capabilities within these two perspectives.

Getting Started with Identity Management

Identity management topics in this chapter include security realm management, security realm group management, role management, user management, and password policy management. In addition, information is provided on restoring user data in the event of Security Server database corruption.

Identity Management Interface

The Security Server provides a command line interface (CLI) and a graphical user interface (GUI) used to perform identity management tasks. This chapter discusses the CLI. (For information on the GUI, see [Chapter 7, “Security GUI.”](#)) The three modes of operation for the CLI are interactive mode, noninteractive mode, and batch mode as explained below:

- Interactive mode provides a prompt-based CLI. Most information in this chapter deals with CLI commands, accessed via the Management Shell (mshell), in interactive prompt-based mode.

To run the Management Shell, log in as the `root` user (using an SSH2 connection if you are accessing remotely), type `mshell` at the command line, and provide your security administrator username/password when prompted.

- Noninteractive mode enables you to automate tasks using a noninteractive scriptable interface.

In noninteractive mode, you can execute the following command at the command line to run the Management Shell, log in, and run the specified command:

```
mshell <username>/<password> <command>
```

The scenario above is what allows for scripts to use noninteractive mode.

As a second scenario for noninteractive mode, you can create a text file of valid CLI commands, with one command on each line. Place the file in a location of your choice. Execute the following command at the command line to run the Management Shell, log in, and run commands in the file:

```
mshell <username>/<password> @<filename>
```

where @<filename> is an absolute path to the file (for example, /tmp/commands.txt)

This second scenario is not meant for scripts.

- Batch mode is a special interactive mode that provides “bulk load” provisioning of security realms, realm groups, roles, and user accounts. See [“Bulk Account Management Operations” on page 31](#).

Identity Management Tasks

After becoming familiar with concepts in the following sections, you can use the references below to go directly to a particular task:

- [“Creating Realms and Realm Groups” on page 17](#)
- [“Modifying Realms and Realm Groups” on page 20](#)
- [“Deleting Realm Groups and Realms” on page 21](#)
- [“Creating Roles” on page 23](#)
- [“Modifying Roles” on page 23](#)
- [“Deleting Roles” on page 23](#)
- [“Finding User Accounts across Realms” on page 24](#)
- [“Viewing User Accounts” on page 25](#)
- [“Viewing Custom Attributes for User Accounts” on page 25](#)
- [“Creating User Accounts” on page 25](#)
- [“Locking/Unlocking User Accounts” on page 26](#)
- [“Modifying User Accounts” on page 27](#)
- [“Deleting User Accounts” on page 28](#)
- [“Viewing Purged Inactive User Accounts” on page 28](#)
- [“Enabling/Disabling User Accounts” on page 29](#)
- [“Restricting User Logins with Login Windows” on page 29](#)
- [“Changing Your Own Password” on page 30](#)
- [“Resetting User Passwords” on page 30](#)

Configuration

Configuration involves setting up security realms, security realm groups, security roles, and user accounts.

Understanding and Managing Security Realms and Realm Groups

The following sections (1) define security realms and realm groups and (2) provide information on creating, modifying, and deleting security realms and realm groups.

What Are Security Realms?

A security realm determines the scope of security data. A realm is the region to which a security ID or permission applies. A user defined as “John” in one realm is treated differently from “John” in a second realm, even if these two “John” IDs represent the same human user.

Security realms in the Comverse ONE solution are usually scoped by application component because many job assignments are typically associated with an application. For example, Product Catalog users are provisioned in one realm and Unified Platform users are provisioned in a separate realm. When defining a security realm, you can also define custom attributes that are part of the realm. All users provisioned for a security realm inherit the realm’s attributes.

What Are Security Realm Groups?

A security realm group provides further scoping (that is, segmentation) of users within a security realm. Security realm groups are used for the following main purposes: (1) to provide common attributes to be associated with a subset of users in the realm and (2) to provide one or more security roles for that group of users. This further segmentation of users within a realm can be useful in certain scenarios where users can have various levels of privileges within a realm. Note the following points about security realm groups:

- A user can belong to multiple groups.
- A group can have custom attributes defined. A user inherits the attributes for all groups that the user belongs to (in addition to inheriting the realm’s attributes, if any).
- A group can have custom values defined for the soft timeout (session inactivity timeout) and hard timeout (maximum session duration timeout) for users in the group.
- Groups determine security roles for users because security roles are associated with groups, not with individual users. A user inherits the roles associated with all groups that the user belongs to. (Through their use in authorization policies, roles specify the application-specific resources/actions that users are authorized to access/perform.)
- A group can have multiple security roles associated with it. If a group does not have any associated custom roles, by default it will be associated with the default roles (ADMIN and GUEST). These default roles exist so that actions by users who belong to the group will not be blocked by lack of a role.
- A DEFAULT group for each security realm is created automatically when the realm is created. No group attributes or security roles can be added to a DEFAULT group (which has only the ADMIN and GUEST default roles). The DEFAULT realm group cannot be deleted.
- Each user must have a priority group assigned from among the groups the user belongs to. The main purpose of this priority group is to resolve potential attribute conflicts. In addition, the priority group establishes the soft timeout and hard timeout values for user login sessions. If a user is defined for a security realm but does not have a priority group assigned, the DEFAULT group for the security realm becomes that user’s priority group (and thus the user inherits the DEFAULT group’s roles and has the soft timeout/hard timeout values defined for that group).

Creating Realms and Realm Groups

The prerequisite step before any users can be provisioned is to create a security realm. After creating a security realm, you can create one or more security realm groups.



NOTE

A user is unique within a realm. Therefore, a user with the same user ID in a different realm is considered a different user.

When creating a security realm, you must provide the *realm name*. It is recommended that you also provide a *realm description* when creating the security realm. You can optionally provide attributes for the realm, which all users provisioned for the realm will inherit. You can also optionally define a customized password policy for the realm that differs from the default password policy if the default policy does not meet your password policy requirements. (Passwords for all user accounts that you later define for a security realm must comply with that realm's password policy.)

Any parts of the password policy not provided during security realm creation are supplied by the default password policy. [Table 4](#) describes the default password policy.

Table 4 Default Password Policy

Parts of the Policy	Default Value	CLI Option
Minimum password length.	6	-plen
Maximum password length.	20	-mxlen
Minimum number of alphabetic characters in the password.	1	-ac
Minimum number of the alphabetic characters that must be lowercase.	0	-al
Minimum number of the alphabetic characters that must be uppercase.	0	-au
Minimum number of other characters in the password, which can be numbers or special characters such as #, &, %, and so on.	1	-oc
Minimum number of characters in a new password that must be different from characters in the old password.	2	-md
Minimum password age, in weeks, before a password can be changed.	0	-mna
Maximum password age, in weeks, at which time the user is notified to change the password.	4	-mxa
Maximum expiration threshold, which is the maximum number of weeks beyond the maximum password age that a password can be changed by the user. (After that, it must be reset by the security administrator.)	2	-mxex
Number of weeks in which a user cannot reuse a previous password.	2	-hiex
Number of previous passwords that a user cannot reuse.	4	-hisz
Maximum number of consecutive failed login attempts (maximum retries) allowed before the user account is locked, disabling logins.	3	-mxr
Lock interval, which is the time span (in minutes) during which consecutive failed login attempts are counted in determining whether to lock a user account.	30	-lkitr
Dictionary list, which is a list of words that are prohibited as passwords. (When specifying a dictionary list, separate the words with commas and enclose the list in quotation marks.) During evaluation of a submitted password, the dictionary list is checked first. If an exact match is found in the dictionary list, no further checking of password constraints occurs and the password cannot be used. If a match is not found in the dictionary list, then the password is evaluated against other parts of the password policy.	None (No list exists in the default password policy.)	-dl

The example in [Figure 1](#) shows how to create a security realm using the Management Shell CLI. In this example, all options for the security realm's password policy are provided except for the minimum number of lowercase and uppercase alphabetic characters. Because those options are not provided, their values will be taken from the default password policy. (This means that, of the minimum number of alphabetic characters required, there is no further requirement related to uppercase or lowercase.) The example also includes definition of two realm attributes.

Figure 1 CLI add_realms Example

```
upm1:root:mshell> add_realms -rlid DEMO_REALM2 -descr DemoRealm2 -plen 9 -mxlen 15 -ac 7
-oc 2 -md 3 -mna 2 -mxa 6 -mxex 6 -hiex 6 -hisz 6 -mxr 4 -lkitr 40 -d1 "Monday,Tuesday"
-att "seven:7,eight:8"
Status Message:
    Realm added successfully
upm1:root:mshell>
```

Once a security realm is created, you can create one or more security realm groups. When creating security realm groups, you must provide the *realm name* that this group will be associated with, and the *group name* to uniquely identify the group within the realm. In addition to the mandatory information for realm groups, you can optionally provide a *realm group description*, one or more custom *attributes* for the group, *role names* that are associated with the group (if security roles have already been created), and a custom *soft timeout* (session inactivity timeout, in minutes) and *hard timeout* (maximum session duration timeout, in minutes) for the group that differ from values in the default session policy. (The default soft timeout is 30 minutes. The default hard timeout is 480 minutes, or 8 hours.)

As previously stated, the purpose of a security realm group is to further segment users within a security realm. The group attributes and the roles associated with the group are what provide this further segmentation. Therefore, if you do not provide group attributes or roles for a security realm group, then essentially you have duplicated the DEFAULT security realm group. When providing multiple attributes for a group, separate the *<name>:<value>* pairs with commas and enclose the group of attributes in quotation marks.



NOTE

When specifying multiple values for an attribute, separate those values with pipe symbols (|).

When providing multiple roles for a group, separate the role names with commas and enclose the group of roles in quotation marks. The example in [Figure 2](#) shows how to create a security realm group using the Management Shell CLI. This example includes definition of two group attributes and two roles associated with the group.

Figure 2 CLI add_group Example

```
upm1:root:mshell> add_group -gid demogrpl -rlid DEMO_REALM1 -descr DemoGroup1
-att "three:3,four:4" -ro "demo_role1,demo_role2"
Status Message:
    Group added successfully
upm1:root:mshell>
```

**NOTE**

Attributes can be defined at the realm, group, and user levels, which means the potential exists for attribute conflicts. See [Appendix B, “Attribute Conflict-Resolution Rules,”](#) for information on how these conflicts are resolved.

Modifying Realms and Realm Groups

Once a security realm has been created, you can add or modify the realm description and short description and add or remove realm attributes. You can also remove the realm’s password policy so that the realm uses the default password policy (see [Table 4, “Default Password Policy,”](#) on [page 18](#)), or you can modify some or all portions of the realm’s password policy.

**CAUTION**

If user accounts have been created for the realm, be cautious in removing the password policy or changing certain portions of the policy, such as minimum password length, minimum number of alphabetic characters, or minimum number of other characters. Otherwise, passwords for user accounts defined for the realm might no longer be valid.

When modifying a realm, you must provide the *realm name* and the information to modify. If you remove realm attributes, be aware that all users defined for the realm will no longer inherit those attributes. If you add realm attributes, all users in the realm will inherit the attributes. To modify portions of the realm’s password policy, specify the *modify* operation and provide the password options to modify and their corresponding values. To remove the realm’s password policy so that the realm reverts to the default password policy, specify the *remove* operation. The example in [Figure 3](#) shows how to modify a realm using the Management Shell CLI. This example illustrates modifying the number of alphabetic characters and other characters in the realm’s password policy and also adding two realm attributes.

Figure 3 CLI modify_realms Example

```
upm1:root:mshell> modify_realms -rlid DEMO_REALM1 -op modify -ac 6 -oc 3
-aa "three:3,four:4"
Status Message:
    Realm updated successfully
upm1:root:mshell>
```

After a security realm group has been created, you can add or modify all details about the realm group except the unique group name and the realm it is associated with. If you remove group attributes, be aware that any users in the group will no longer inherit those attributes. If you add group attributes, all users in the group will inherit the attributes. Similarly, if you add roles to the group, all users in the group will inherit the roles. If you remove roles from the group, all users in the group will no longer inherit the roles.

**NOTE**

Attributes can be defined at the realm, group, and user levels, which means the potential exists for attribute conflicts. See [Appendix B, "Attribute Conflict-Resolution Rules,"](#) for information on how these conflicts are resolved.

When modifying a security realm group, you must provide the *group name*, the *realm name* it is associated with, and the information to modify. The example in [Figure 4](#) shows how to modify a realm group using the Management Shell CLI. This example illustrates adding two group attributes and two roles.

Figure 4 CLI modify_group Example

```
upm1:root:mshell> modify_group -gid demogrp1 -rlid DEMO_REALM1 -aa "five:5,six:6"
-ar "demo_role3,demo_role4"
Status Message:
    Group updated successfully
upm1:root:mshell>
```

Deleting Realm Groups and Realms

**NOTE**

Before deleting a group that is the priority group for any users, it is recommended that you reassign those users to a different priority group.

Deleting a security realm group means that any group attributes and roles, which were previously inherited by users in the group, will no longer be inherited by those users. If the deleted group was the priority group for any users, those users will be reassigned to the DEFAULT group as their priority group. (The users will then inherit the DEFAULT group's roles, which are ADMIN and GUEST, and the soft timeout/hard timeout values of the DEFAULT group.)

When deleting a security realm group, you must provide the *group name* and the security *realm name* that the group is associated with. The example in [Figure 5](#) shows how to delete a security realm group using the Management Shell CLI.

Figure 5 CLI remove_group Example

```
upm1:root:mshell> remove_group -gid demogrp1 -rlid DEMO_REALM4
Status Message:
    Group removed successfully
upm1:root:mshell>
```

When deleting a security realm (*see the Warning after the next figure*), you must provide the *realm name*. The example in [Figure 6 on page 22](#) shows how to delete a security realm using the Management Shell CLI.

Figure 6 CLI remove_realms Example

```
upml:root:mshell> remove_realms -rlid DEMO_REALM4
Status Message:
    Realm removed successfully

upml:root:mshell>
```



Deleting a security realm deletes all user accounts and realm groups defined for the realm, along with the realm itself. Carefully consider whether you want to delete a security realm. (Deleting security realms does not delete roles because roles are independent of realms and realm groups and have only an *association* with realm groups.)



The UPSEC security realm cannot be deleted. This realm contains the administrator user and users who can perform actions on the Security Server and Unified Platform. Deletion of this realm is not allowed because it would destroy the administrator account.

Understanding and Managing Security Roles

The following sections (1) define security roles and (2) provide information on creating, modifying, and deleting security roles.

What Are Security Roles?

A security role is a privilege granted to groups of users based on specific conditions. By its association with a realm group, a security role allows you to define access to resources for multiple users at once, based on their realm group membership. Granting a security role to a group confers the defined access privileges on all users in that group (via the role's use in authorization policies). For example, you might define a security role called `AppAdmin` that has write access to a particular application's resources. Users in any group granted the `AppAdmin` security role would then have write access to those resources. Multiple groups can be granted a single security role.



Be aware that authorization policies must be defined to specify the application-specific resources/actions that users in a specific security role are authorized to access/perform. It is important to understand that a role assumes meaning only within the context of authorization policies and does not by itself confer any access privileges. Instead, rules in authorization policies specify which subjects (currently, roles) can perform which actions on which resources.

Creating Roles

**NOTE**

Assigning roles to groups is not done during role creation. Instead, it is done during group creation or group modification.

When creating a security role, you must provide the unique *role name*. It is recommended that you also provide a *role description*. The example in [Figure 7](#) shows how to create a security role using the Management Shell CLI.

Figure 7 CLI add_role Example

```
upm1:root:mshell> add_role -roid demo_role1 -descr DemoRole1
Status Message:
    Role added successfully
upm1:root:mshell>
```

**NOTE**

When defining security roles, you must understand that a security role is independent of a security realm. This implies that the security role called ADMIN, which is created during database installation, can be bound to realm groups of multiple security realms. Therefore, you are not required to create an ADMIN security role for each realm. How the security role is used within each security realm can differ from realm to realm within the Security Server.

Once a security role is created, it can then be used when defining authorization policies to specify the application-specific resources/actions that users in groups associated with this security role are authorized to access/perform.

Modifying Roles

After a security role is created, you cannot modify it. The reason for this restriction is to preserve the role's association with groups. If the role is not associated with any groups and you want to change the role name, for example, just delete the role and add a new role, specifying the desired name.

Deleting Roles

Be aware that deleting a security role removes any current associations with groups, thus removing access privileges provided by the role for users in those groups. Deleting a role produces this result because authorization policies that define access to application-specific resources are based on roles. Also be aware that if a group is associated with only one role, and that role is deleted, then the group is associated with the default roles (ADMIN and GUEST).

**NOTE**

The ADMIN role cannot be deleted because that role is for the security administrator and other users with administrative privileges.

When deleting a role, you must provide the unique *role name*. The example in [Figure 8](#) shows how to delete a security role using the Management Shell CLI.

Figure 8 CLI remove_role Example

```
upm1:root:mshell> remove_role -roid DEMO_ROLE2
Status Message:
    Role removed successfully

upm1:root:mshell>
```

Understanding and Managing Password Policies

As previously explained, password policies are defined at the security realm level. Passwords for all user accounts defined for a security realm must comply with that realm's password policy. If you do not specify any password policy options when creating a security realm (and do not modify the realm to provide the options), then the default password policy is used for that realm. Similarly, if you specify only selected password policy options (but not all options) when creating a realm, the default password policy supplies values for those password policy options not specified. For details about the default password policy, see [Table 4, "Default Password Policy," on page 18](#).

Working with User Accounts

The following sections discuss (1) how to work with accounts from the Management Shell CLI and (2) how to perform bulk load operations for user provisioning with a batch process.

User Account Management

The following sections discuss how to work with user accounts from the Management Shell CLI. For information on using a batch process for bulk loading of user accounts and associated entities such as realms, realm groups, and roles, see ["Bulk Account Management Operations" on page 31](#).

Finding User Accounts across Realms

User accounts are defined per security realm, and all operations on a user account occur within the context of the realm in which it is defined. It is possible to search across realms to find a user account. Among other information, the displayed output of the search shows the realm where the user account is defined. To search across realms, provide one or more of the following: *user ID*, *first name* of the user, *last name* of the user, *email address* of the user, and *lock status* of the user account (true/false). The example in [Figure 9](#) shows how to find a user account across realms using the Management Shell CLI. In this example, the ignore option excludes several columns of information (such as MiddleName, Department, and so on) from the displayed output.

Figure 9 CLI find_users Example

```
upm1:root:mshell> find_users -ln jones -fn fred -i MiddleName,Department,Extension,Phone,
CreatedDate,LastLogin,Email,ForcePasswordChange,LockStatus

User Information
UserId      FirstName  LastName  RealmName  PriorityGroup  Groups          UserAttributes
fjones01    Fred      Jones     DEMO_REALM1  DEMOGRP2      [DEMOGRP2] {one:1,two:2,five:5,six:6}

upm1:root:mshell>
```

Viewing User Accounts

User accounts are defined per security realm, and you can view one or all user accounts in a realm. To view all user accounts, you must provide the *realm name*. To view an individual account, you must provide the *user ID* to uniquely identify the user and the *realm name*. For simplified viewing, you can optionally specify parts of the user information to ignore, which means they are not displayed. The example in [Figure 10](#) shows how to view an individual account using the Management Shell CLI. In the example, the ignore option excludes several columns of information (such as MiddleName, Department, and so on) from the displayed output. Note that the UserAttributes column includes attributes specifically defined for the user, attributes inherited from the security realm, and attributes inherited from all groups that the user belongs to.

Figure 10 CLI list_users Example

```
upm1:root:mshell> list_users -rlid DEMO_REALM1 -uid jsmith01 -i MiddleName,Department,Extension,Phone,Email,
LastLogin,LockStatus,ForcePasswordChange,RealmName,CreateDate,AccountState

User Information
UserId   FirstName  LastName  PriorityGroup  Groups                                UserAttributes
jsmith01 John       Smith     demogrp1      [DEMOGRP2, DEMOGRP1] {one:1,two:2,five:5,three:3,four:4,six:6}

upm1:root:mshell>
```

Viewing Custom Attributes for User Accounts

You can view the custom attributes for all users, or an individual user, within a security realm. The displayed output indicates whether custom attributes are inherited from the realm or group levels, or are defined at the individual user level. To view custom attributes for users, you must provide the *realm name*. You can also optionally provide the *user ID*. The example in [Figure 11](#) shows how to view custom attributes using the Management Shell CLI.

Figure 11 CLI list_attributes Example

```
upm1:root:mshell> list_attributes -rlid DEMO_REALM1

Attributes Information
UserId   RealmName  RealmAttributes  GroupAttributes  UserAttributes
jsmith01 DEMO_REALM1 {one:1,two:2}    {five:5,four:4,three:3,six:6} {eight:8,seven:7}
fjones01 DEMO_REALM1 {one:1,two:2}    {five:5,six:6}    {}

upm1:root:mshell>
```

Creating User Accounts

User accounts are defined per security realm, so you can create user accounts only after creating security realms. Each user must be assigned to at least one realm group and to a priority group from among those groups. This assignment can be done when creating the user account (if groups have already been created) or later by modifying the user account.



NOTE

A user is unique within a realm. Therefore, a user with the same user ID in different realms is considered a different user.

When creating user accounts, you must provide the following information: a unique *user ID* within the security realm, the *realm name* in which you are creating the user account, the user's *first name*, the user's *last name*, and the user's *password*. The password must comply with the realm's password policy.

In addition to the mandatory information, you can optionally provide the user's *middle name*, *phone number*, *phone extension*, *department*, *email address*, one or more *group names* within the security realm (if the groups have already been created), a *priority group* from among the specified groups,

and *user attributes* (custom attributes). You can also optionally specify the account *lock* status (true/false), whether to *force change password* at the initial login (true/false), and the account state (ENABLED/DISABLED). User attributes define application-specific custom attributes associated with the user, such as reseller ID and dealer ID. When specifying multiple user attributes, separate the `<name>:<value>` pairs for attributes with commas and enclose the entire group of attributes in quotation marks.

**NOTE**

When specifying multiple values for an attribute, separate those values with pipe symbols (|).

**NOTE**

Attributes can be defined at the realm, group, and user levels, which means the potential exists for attribute conflicts. See [Appendix B, “Attribute Conflict-Resolution Rules.”](#) for information on how these conflicts are resolved.

If you do not lock the user account at creation, by default the account is created unlocked. Locking the account at creation prevents user logins until you unlock the account. If you do not indicate whether a password change at initial login is required, by default the user is not forced to change the password. If you do not provide the account state at creation, by default the state is ENABLED, meaning that the user account will be purged after a certain number of days of inactivity. (See [“Viewing Purged Inactive User Accounts” on page 28](#) for more information.) The example in [Figure 12](#) shows how to create a user account using the Management Shell CLI. This example illustrates locking the account at creation, indicates that the user will be forced to change the password at initial login, specifies group membership in two groups, and assigns the user’s priority group.

Figure 12 CLI add_user Example

```
upm1:root:mshell> add_user -uid jsmith01 -rlid DEMO_REALM1 -fn John -ln Smith
-pwd jjhs#s@hh -gid "demogrp1,demogrp2" -lck true -fcp true -pgroup demogrp1
Status Message:
    User created successfully
upm1:root:mshell>
```

Locking/Unlocking User Accounts

As previously discussed, locking a user account temporarily disables logins for the user, and unlocking the account enables logins. A user account is locked automatically under the following circumstances:

- The account was created with a locked status and has not been unlocked.
- The user has exceeded the maximum number of consecutive failed login attempts (maximum retries) within the time span (lock interval) defined in the security realm’s password policy. You must unlock the account manually.
- The user has failed to change the password before the maximum expiration threshold defined in the security realm’s password policy. The account remains locked until you reset the password and unlock the account.

To lock or unlock an existing user account, two methods are available:

- Use the `lock_user` or `unlock_user` commands. You must provide the unique *user ID* and the security *realm name* where the user account is defined. The example in [Figure 13](#) shows how to lock and unlock a user account with these commands using the Management Shell CLI.
- Modify the account, providing the unique *user ID*, the security *realm name* where the user account is defined, and the desired *lock status* (true/false). The example in [Figure 14 on page 28](#) shows how to modify a user account, including unlocking the account, using the Management Shell CLI.

Figure 13 CLI `lock_user` and `unlock_user` Examples

```
upm1:root:mshell> lock_user -uid jsmith -rlid DEMO_REALM1
Status Message:
    User updated successfully

upm1:root:mshell> unlock_user -uid jsmith -rlid DEMO_REALM1
Status Message:
    User updated successfully

upm1:root:mshell>
```

Modifying User Accounts

You can modify user accounts by adding or changing information. When modifying a user account, you must provide the unique *user ID*, the security *realm name* where the user account is defined, and the information to modify. (You cannot change a user's password by modifying the user account. Instead, you must reset the password. See [“Resetting User Passwords” on page 30](#). Changing your own password requires a different method. See [“Changing Your Own Password” on page 30](#).) Note the following points about adding/removing group associations for a user account:

- If you remove all custom group associations for a user, the user is assigned to the DEFAULT group for the security realm, and that group becomes the user's priority group. Therefore, the user inherits the DEFAULT group's roles (ADMIN and GUEST) and has the soft timeout/hard timeout values defined for the DEFAULT group.
- If you remove a group that is assigned as the user's priority group and do not assign another priority group, the DEFAULT group for the security realm is assigned as that user's priority group. This assignment can be changed later.
- You cannot manually create an association with the DEFAULT realm group for a user, but you can remove that group association.
- If you attempt to remove the DEFAULT realm group as a group association for the user, and the DEFAULT group is set as the user's priority group, an error will be displayed.

The example in [Figure 14 on page 28](#) shows how to modify a user account using the Management Shell CLI. This example illustrates adding the user's department, unlocking the account, removing a group association that is the user's current priority group, assigning a different priority group, and adding two user attributes.

Figure 14 CLI modify_user Example

```
upm1:root:mshell> modify_user -rlid DEMO_REALM1 -uid jsmith01 -dept CustServ  
-lock false -rg demogrp1 -pgroup demogroup2 -aa "three:3,four:4"  
Status Message:  
    User updated successfully  
upm1:root:mshell>
```

**NOTE**

Attributes can be defined at the realm, group, and user levels, which means the potential exists for attribute conflicts. See [Appendix B, "Attribute Conflict-Resolution Rules,"](#) for information on how these conflicts are resolved.

Deleting User Accounts

When deleting a user account, you must provide the security *realm name* where the user account is defined and the unique *user ID*. Deleting a user account also deletes the user association with groups. The example in [Figure 15](#) shows how to delete a user account using the Management Shell CLI.

Figure 15 CLI remove_user Example

```
upm1:root:mshell> remove_user -rlid DEMO_REALM1 -uid fjones01  
Status Message:  
    User removed successfully  
upm1:root:mshell>
```

**NOTE**

The `secadmin` user account cannot be deleted. That is the account for the security administrator.

Viewing Purged Inactive User Accounts

After being inactive for a specified number of days (the default is 30 days), a user account is purged if its account state is `ENABLED` and the user is not a system user. (A system user has the `usertype` user-level attribute defined with a value of `system`. See ["Well-Known Attributes" on page 191](#).)

In this context, the term "purged" means that the record is deleted from the `SEC_IDM_USER` database table and inserted in the `SEC_IDM_PURGED_USERS` table. A background purge job is scheduled to run once a day to purge appropriate user accounts. After being purged, a user account is no longer displayed with the `list_users` command. The example in [Figure 16 on page 29](#) shows how to view purged user accounts using the Management Shell CLI.

Figure 16 CLI `list_purged_users` Example

```
upm1:root:mshell> list_purged_users
```

A job is scheduled to run once a week (that is, every Sunday) against the `SEC_IDM_PURGED_USERS` table. If the number of records in the table is greater than the configured allowable number (the default is 50), all records in the table are deleted. At this point, these purged user accounts can no longer be viewed.

**NOTE**

Two configurable system properties specify (1) the number of days a user account can remain inactive before being purged and (2) the number of purged inactive user records allowed in the `SEC_IDM_PURGED_USERS` table:
`user.idletime` = days an account can remain inactive
`idleusers.count` = number of purged inactive records allowed
These properties are located in the following file:
`$JBOSS_HOME/conf/application.properties`

Enabling/Disabling User Accounts

As discussed in [“Viewing Purged Inactive User Accounts” on page 28](#), the account state determines whether a user account will be purged after a certain number of days of inactivity, as long as it is not a system user account. (An account state of `ENABLED` means the account will be purged, and `DISABLED` means it will not be purged.) To enable/disable a user account state, you must provide the unique *user ID* and the security *realm name* where the user account is defined. The examples in [Figure 17](#) show how to disable and enable a user account using the Management Shell CLI.

Figure 17 CLI `disable_user` and `enable_user` Examples

```
upm1:root:mshell> disable_user -uid jsmith -rlid DEMO_REALM1
Status Message:
    User updated successfully

upm1:root:mshell> enable_user -uid jsmith -rlid DEMO_REALM1
Status Message:
    User updated successfully

upm1:root:mshell>
```

Restricting User Logins with Login Windows

You might need to ensure that certain users can log in only during specific times, such as in a time window between 8:00 A.M. and 5:00 P.M., Monday through Friday. To establish that restriction, use the `loginwindow` attribute, which can be defined at the security realm, group, or user level. The value for the attribute is a `cron` expression that controls when logins are permitted. (For general information about `cron` expressions, see [Appendix D, “Cron Expressions.”](#)) Multiple `cron` expressions for the attribute, separated by pipe (`|`) symbols, are supported.

The example in [Figure 18 on page 30](#) shows how to modify a group to add the `loginwindow` attribute. Adding the attribute at the group level means that the login window applies to all users who belong to that group. In this example, the login window is anytime (literally, any second) between 8:00 A.M. and 5:00 P.M., Monday through Friday.

Figure 18 CLI modify_group Example to Define a Login Window

```
upm1:root:mshell> modify_group -rlid DEMO_REALM1 -gid demo_group1  
-aa loginwindow:"* * 8-17 ? * MON-FRI"  
Status Message:  
    Group updated successfully  
upm1:root:mshell>
```

The value in the Hours field of the `cron` expression cannot have overlapping hours (that is, the hours cannot span into the next day). For example, 20-4 in the Hours field does not work. To handle this issue, you can provide multiple `cron` expressions for the `loginwindow` attribute, separated by pipe (|) symbols. For example:

```
loginwindow:"* * 20-23 ? * MON-FRI|* * 0-4 ? * TUE-SAT"
```

**NOTE**

Login window definitions should be in accordance with your company's security policies. Incorrectly defined login windows can prevent authorized users from accessing the system during authorized working hours.

Password Management

Using the Management Shell CLI, you can change your own password or reset user passwords. The following sections provide details.

Changing Your Own Password

After logging in to the Management Shell, you can change your own password. To do this, you must provide the *previous password* (that is, the password currently in use) and the *new password*. The new password must comply with the security realm's password policy. The example in [Figure 19](#) shows how to change your password using the Management Shell CLI.

Figure 19 CLI change_password Example

```
upm1:root:mshell> change_password -prevpass abc#123 -newpass pqrs*234
```

The `change_password` command is used only for changing the password of the currently logged in Management Shell (`mshell`) user. See [“Resetting User Passwords”](#) below for information on changing passwords for other users.

Resetting User Passwords

Resetting a user password changes it to a randomly generated password that complies with the security realm's password policy. After a password reset, the user is required to change the password at the next login. When resetting a user password, you must provide the security *realm name* where the user account is defined and the unique *user ID* within the realm. The example in [Figure 20 on page 31](#) shows how to reset a user password using the Management Shell CLI.

Figure 20 CLI reset_password Example

```
upm1:root:mshell> reset_password -uid pjenki01 -rlid DEMO_REALM2
Status Message:
    Reset password successful
New Password : 87gUz
upm1:root:mshell>
```

Bulk Account Management Operations

Instead of using the CLI's prompt-based mode to create security realms, realm groups, roles, and user accounts, you can use a batch process for a bulk load operation. The input file for the batch process is a Microsoft Excel worksheet (also called a spreadsheet).

**NOTE**

For bulk load operations, the 2007 Office Open XML format is not supported. This means that spreadsheets must be saved in the 2003 format (that is, with the .xls extension instead of the .xlsx extension).

A sample worksheet template, named `IDMAdministrationTemplate.xls`, is installed during Security Server installation and is located in `$JBOSS_HOME/templates/security`. For the Converse ONE solution, the security realms align with applications. Some illustrative examples can be found in the sample worksheet template.

As explained in the worksheet template, the following three styles of provisioning are available:

- **Basic:** Use this style of provisioning when multiple groups are not required, and all users will belong to the DEFAULT group. This style, unlike the Normal provisioning style discussed below, does not allow for group-level or user-level custom attributes. The Basic style allows for only standard user attributes to be provisioned.
- **Normal:** Use this style of provisioning when multiple groups are not required, and all users will belong to the DEFAULT group. This style allows for user-level attributes to be defined, but not group-level attributes, as there are no customized groups for the given realm.
- **Advanced:** Use this style of provisioning when there are multiple groups within a given realm. Users are assigned to various groups within the realm. Each group can have one or more group-level custom attributes that all users in that group will inherit. Also, at the user level, custom attributes can be assigned. This style is the most complex as it allows you to provision custom groups with custom group-level attributes, and provision users to each group as well as provide user-level custom attributes.

For the bulk load operation, do the following five steps:

1. Using the worksheet template as a guide, create a new worksheet based on your chosen provisioning style.
2. Complete the worksheet, following the instructions and comments provided in the template.
3. Save your worksheet file with a descriptive name, indicating the appropriate application (for example, `CSM_Idm.xls`).
4. Place the completed worksheet on the Security Server in either the default location (`$JBOSS_HOME/batch/users` directory) or in a directory of your choice.

5. To process the worksheet for the bulk load operation, do one of the following steps:
 - a. If the worksheet is located in the default location, execute the following command using the Management Shell CLI:
add_user -b
This command processes all worksheet files in the `$JBOSS_HOME/batch/users` directory and loads the data into the database.
 - b. If the worksheet is located in another directory, specify the directory in the command. For example, if the directory is `/tmp`, execute the following command using the Management Shell CLI:
add_user -b /tmp
This command processes all worksheet files in the `/tmp` directory and loads the data into the database.

Worksheet files remain in the directory where they were placed unless you manually remove them.

Chapter 4

Policy Management

4

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To



Policy Management Overview

Policies are used to make decisions about authorization. Authorization refers to the process of giving permission to perform an action on an application resource based on some authenticated identity, such as a role. In practical terms, a policy is a set of rules defining which subjects are permitted to access which resources using which actions under which conditions.

Policies conform to version 2.0 of the Extensible Access Control Markup Language (XACML) standard whose governing body is the Organization for the Advancement of Structured Information Standards (OASIS). Information is available from this website:

http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml.

What Is XACML?

XACML is an XML-based language that specifies schemas for authorization policies and for authorization decision requests and responses. It also specifies how to evaluate policies against requests to compute a response. Expressed another way, XACML provides the following:

- **Policy Control Language:** The policy language is used to define authorization policies (also known as XACML policies).
- **Request/Response Language:** The request/response language expresses queries about whether a particular type of access should be allowed (requests) and describes answers to those queries (responses).
- **Runtime Architectural Components:** The security platform for the Converse ONE solution makes use of the following XACML runtime components: Policy Enforcement Point (PEP), Policy Decision Point (PDP), and Policy Administration Point (PAP). These components work together to either permit or deny requests for access to resources. Later sections in this chapter discuss these components in more detail.

XACML Policies

It is outside the scope of this document to provide comprehensive information about XACML policies. The following is a general description to provide some background information.

An XACML policy consists of these major elements: (1) a target, (2) a set of rules, (3) an identifier for rule-combining algorithms, and (4) a set of obligations.

- **Target:** The policy target. Each policy has only one target. The target helps to determine whether the policy is relevant for the request. The policy's relevance to the request determines if the policy is to be evaluated for that request. This determination is achieved by defining attributes of the following three categories in the target, along with their values:
 - Subject
 - Resource
 - Action
- It is not mandatory to have attributes for all three categories in a target. The values of these attributes are compared with the values of the same attributes in the request. If they match (after applying some function on the attributes), the policy is considered relevant to the request and is evaluated.
- **Rules:** Multiple rules can be associated with a policy. Each rule is composed of a condition, an effect, and a target.
 - **Condition:** Condition is a statement about attributes that, on evaluation, returns True, False, or Indeterminate.
 - **Effect:** Effect is the intended consequence of the satisfied rule. It can take the value Permit or Deny.

- **Target:** The rule target, as in the case of the policy target discussed previously, helps to determine whether or not a rule is relevant for a request. The mechanism for achieving this determination is also similar to how it is done for the policy target. That is, this determination is achieved by defining attributes in the following three categories in the target, along with the attribute values:

- Subject
- Resource
- Action

The final outcome of the rule depends on the condition evaluation. If the condition returns `Indeterminate`, the rule also returns `Indeterminate`. If the condition returns `False`, the rule returns `NotApplicable`. If the condition returns `True`, the value of `Effect` is returned, which is either `Permit` or `Deny`.

- **Rule-Combining Algorithm:** As previously explained, a policy can have multiple rules. It is possible for different rules to generate conflicting results, and rule-combining algorithms resolve such conflicts to arrive at one outcome per policy per request. Only one rule-combining algorithm applies per policy. The algorithms are: deny-overrides, ordered-deny-overrides, permit-overrides, ordered-permit-overrides, or first-applicable.
 - **deny-overrides:** If any rule evaluates to `Deny`, the final authorization decision is also `Deny`.
 - **ordered-deny-overrides:** The same as deny-overrides, except that the order in which relevant rules are evaluated is the same as the order in which they are added in the policy.
 - **permit-overrides:** If any rule evaluates to `Permit`, the final authorization decision is also `Permit`.
 - **ordered-permit-overrides:** The same as permit-overrides, except that the order in which relevant rules are evaluated is the same as the order in which they are added in the policy.
 - **first-applicable:** The result of the first relevant rule encountered is the final authorization decision.
- **Obligations:** One of the objectives of XACML is to provide much finer-level access control than mere permit and deny decisions. Obligations, an optional element in a policy, provide the mechanism for achieving this fine-level control. Obligations are the actions that must be performed by the Policy Enforcement Point (PEP) in conjunction with enforcement of an authorization decision.

Policy Enforcement Point

The Policy Enforcement Point (PEP) is the component that makes requests for authorization and enforces authorization decisions. In the Comverse ONE solution, the Unified API (sometimes referred to as the Single API, or SAPI) is an example of a PEP.

When a subject attempts to perform an action on a resource (for example, to modify account data), the request for authorization to perform the action initially goes to the PEP. The PEP creates an XACML request, sends it to the Policy Decision Point (PDP), and then acts on the PDP's decision. That is, the PEP either permits or denies the request. The PEP also executes any policy obligations that need to be performed.

Policy Decision Point

The Policy Decision Point (PDP) is the component that makes decisions about authorization requests. A PDP can be deployed in multiple ways, including embedded in an application if required for performance reasons. At initialization, the PDP loads the policy configuration that defines which policies it manages. Policies are retrieved from the Security Server by the Uniform

Platform Agent (UPA) requesting the policies for a given node type (such as SAPI). This request occurs over a secure channel. Once the policies are retrieved, they are loaded by the PDP service hosted on the UPA.

The PDP receives an authorization request from the PEP. The PDP evaluates the request against the policies that it manages and then returns an XACML response (that is, an authorization decision) to the PEP. It also returns information on any applicable policy obligations.

Policy Administration Point

The Policy Administration Point (PAP) is the component that creates policies and stores them in a repository. In the Comverse ONE solution, the Security Server acts as the PAP.

Getting Started with Policy Management

Topics in this chapter provide information on authorization rules and authorization policies.

Policy Management Interface

The Security Server provides a command line interface (CLI) and a graphical user interface (GUI) used to perform policy management tasks. This chapter discusses the CLI. (For information on the GUI, see [Chapter 7, “Security GUI.”](#)) The three modes of operation for the CLI are interactive mode, noninteractive mode, and batch mode as explained below:

- Interactive mode provides a prompt-based CLI. Most information in this chapter deals with CLI commands, accessed via the Management Shell (mshell), in interactive prompt-based mode.

To run the Management Shell, log in as the `root` user (using an SSH2 connection if you are accessing remotely), type `mshell` at the command line, and provide your security administrator username/password when prompted.

- Noninteractive mode enables you to automate tasks using a noninteractive scriptable interface.

In noninteractive mode, you can execute the following command at the command line to run the Management Shell, log in, and run the specified command:

```
mshell <username>/<password> <command>
```

The scenario above is what allows for scripts to use noninteractive mode.

As a second scenario for noninteractive mode, you can create a text file of valid CLI commands, with one command on each line. Place the file in a location of your choice. Execute the following command at the command line to run the Management Shell, log in, and run commands in the file:

```
mshell <username>/<password> @<filename>
```

where `@<filename>` is an absolute path to the file (for example, `/tmp/commands.txt`)

This second scenario is not meant for scripts.

- Batch mode is a special interactive mode that provides “bulk load” operations for authorization policies and associated rules. See [“Bulk Policy Operations” on page 45](#).

Policy Management Tasks

After becoming familiar with concepts in the following sections, you can use the references below to go directly to a particular task:

- [“Viewing Rules” on page 38](#)
- [“Creating Rules” on page 39](#)

- [“Modifying Rules” on page 39](#)
- [“Deleting Rules” on page 40](#)
- [“Viewing Policies” on page 41](#)
- [“Creating Policies” on page 42](#)
- [“Modifying Policies” on page 42](#)
- [“Deleting Policies” on page 43](#)
- [“Publishing Policies” on page 43](#)
- [“Resynchronizing Policies” on page 44](#)

Working with Rules

An authorization rule is the most elementary unit of an authorization policy.

See [“Rule Management”](#) below for information on using the Management Shell CLI to work with authorization rules. See [“Bulk Policy Operations” on page 45](#) for information on using a batch process to create policies, including their rules, in a bulk load operation. Bulk policy loading is the recommended approach if policies have many rules.

Rule Management

The following sections provide details on viewing, creating, modifying, and deleting authorization rules using the Management Shell CLI. When creating an authorization policy, you must associate one or more rules with that policy. Therefore, rules must be created before policies can be created.



NOTE

There is no one-to-one mapping of rule to policy. That is, a rule is independent of any given policy, and the same rule can be used in multiple policies.

Viewing Rules

You can view all authorization rules defined in the Security Server database, an individual rule, or a group of rules with a common prefix in their names. (For example, if multiple rules are named *ABC<SOMETHING>*, *ABC* is the prefix.) Alternatively, you can view the list of rules associated with an individual authorization policy or a group of policies with a common prefix in their names.

When viewing an individual authorization rule, you must provide the unique *rule name*. When viewing rules with a common prefix in the rule names, you must provide the *prefix*. When viewing the list of rules associated with an individual authorization policy or a group of policies with a common prefix, you must provide the unique *policy name* or the *prefix* in the policy names. The examples in [Figure 21 on page 39](#) show how to view the list of rules associated with an individual authorization policy, and then how to view the details of an individual rule for that policy, using the Management Shell CLI.

Figure 21 CLI list_auth_rule Example

```

upm1:root:mshell> list_auth_rule -pid UP_DEFAULT_ACCESS
Rule Information

PolicyId                RuleId
UP_DEFAULT_ACCESS       PERMIT_ADMIN
UP_DEFAULT_ACCESS       DENY_ALL

upm1:root:mshell> list_auth_rule -id PERMIT_ADMIN
Rule Information

Id          Effect      Subject    Action    Resource
PERMIT_ADMIN Permit      ADMIN      ANY       ANY

upm1:root:mshell>

```

The following information describes details for the rule named PERMIT_ADMIN shown in [Figure 21](#). For Effect, Permit represents the consequence of the rule. For Subject, ADMIN is the administrator role. For Action, ANY means any action (such as create, modify, delete, invoke, enable, disable, and so on). For Resource, ANY means any resource. (Resources can be data, service, or system components. For example, service methods can be resources.) So the PERMIT_ADMIN rule shown in [Figure 21](#) means that users in the ADMIN role are permitted to perform any action on any resource in the system. In this case the term “system” refers to the Unified Platform (UP) because the rule is associated with a policy for the UP.

Creating Rules

When creating an authorization rule, you must provide a *rule name* to uniquely identify the rule within the Security Server. It is also recommended that you provide a *rule description*. You can optionally provide one or more *subjects* (currently, these are roles), *resources*, and *actions* on resources that identify what the rule applies to (that is, the attributes that specify the rule’s target). The resources and types of actions are application-specific. If you do not provide a subject, resource, or action, the default is Any. For multiple subjects, resources, or actions, separate the items with commas and enclose them in quotation marks. You can also optionally provide an *effect* (Permit or Deny) to identify the intended consequence of a satisfied rule. If you do not provide an effect, the default is Permit. (Conditions as part of the rule are not supported through the CLI.) The example in [Figure 22](#) shows how to create an authorization rule using the Management Shell CLI.

Figure 22 CLI create_auth_rule Example

```

upm1:root:mshell> create_auth_rule -id PERMIT_ALL_DEMO -description
"Permit all to DemoRole1" -subject DemoRole1 -effect Permit
Status Message:
    Successfully added rule PERMIT_ALL_DEMO

upm1:root:mshell>

```

Once authorization rules are created, you can use them in defining authorization policies.

Modifying Rules

After an authorization rule has been created, you can modify the description, subjects, resources, actions, or effect.

When modifying an authorization rule, you must provide the *rule name* and the information you want to modify. The example in [Figure 23](#) shows how to modify an authorization rule using the Management Shell CLI. This example illustrates changing the rule description and subject.

Figure 23 CLI modify_auth_rule Example

```
upm1:root:mshell> modify_auth_rule -id PERMIT_ALL_DEMO -description  
"Permit all to DemoRole2" -subject DemoRole2  
Status Message:  
    Successfully updated rule PERMIT_ALL_DEMO  
upm1:root:mshell>
```



NOTE

Modification of an authorization rule does not affect published authorization policies that currently use the rule. (Applications use policies only after they have been published.) Policies must be republished before changes to their rules take effect for target applications. See [“Publishing Policies” on page 43](#). Note that changing a rule description does not require republishing the policy.

Deleting Rules

You can delete an individual authorization rule or a group of rules with a common prefix in their names. (For example, if multiple rules are named *ABC<SOMETHING>*, *ABC* is the prefix.) Deleting an authorization rule also deletes the rule from any policies it is associated with. When deleting rules, you must provide the unique *rule name* or the *prefix* in the rule names. If you provide a prefix, all rules with a matching prefix are deleted. The example in [Figure 24](#) shows how to delete an individual authorization rule using the Management Shell CLI.

Figure 24 CLI remove_auth_rule Example

```
upm1:root:mshell> remove_auth_rule -id PERMIT_ALL_DEMO  
Status Message:  
    Successfully deleted 1 rules matching PERMIT_ALL_DEMO  
upm1:root:mshell>
```



NOTE

Deletion of a rule does not affect published policies that currently use the rule. Policies must be republished before a rule deletion takes effect. See [“Publishing Policies” on page 43](#).

Bulk Rule Operations

See [“Bulk Policy Operations” on page 45](#) for information on creating policies, including their rules, in a bulk operation via a batch process.

Working with Policies

See [“Policy Management”](#) below for information on using the Management Shell CLI to view, create, modify, and delete authorization policies. See [“Bulk Policy Operations” on page 45](#) for information on using a batch process to create policies in a bulk load operation. Bulk policy loading is the recommended approach for loading many policies or policies with many rules.

Policy Management

The following sections provide details about working with authorization policies using the Management Shell CLI. Note that commands provided by the Management Shell CLI support creating basic Security Server policies and, therefore, require a security realm as the entry point of a policy (that is, as the policy target). The rules associated with the policies specify the subject(s), resource(s), and action(s).



NOTE

The Comverse security solution supports the full XACML specification, but the interface that allows for policy creation has been simplified to expose only the XACML properties that are required for Comverse ONE components.

If more advanced XACML policies are required, they must be defined outside the Security Server interface in the XACML native format. For more information, see [“Manual Policy Creation” on page 46](#).

Viewing Policies

You can view details of all authorization policies in the Security Server database, an individual policy, or a group of policies with a common prefix in the policy names. (For example, if multiple policies are named *ABC<SOMETHING>*, *ABC* is the prefix.) When viewing an individual authorization policy, you must provide the unique *policy name*. When viewing policies with a common prefix in policy names, you must provide the *prefix*. The example in [Figure 25](#) shows how to view an individual authorization policy using the Management Shell CLI

Figure 25 CLI list_auth_policy Example.

```
upm1:root:mshell> list_auth_policy -id UP_DEFAULT_ACCESS
Policy Information
```

Id	Realm	CombiningAlg	Number of Rules
UP_DEFAULT_ACCESS	UPSEC	PERMIT-OVERRIDES	2

```
upm1:root:mshell>
```

As shown in [Figure 25](#), viewing a policy shows the number of authorization rules associated with that policy. To see the rule names, use the `list_auth_rule` command and specify the policy name with the `-pid` option. For more information, see [“Viewing Rules” on page 38](#).

Creating Policies



NOTE

When naming policies, identify the realm to which the policy applies by using the realm name as the first few characters of the policy name. Two examples of policy names are PC_<SomeName> and SAPI_<SomeName>, where PC and SAPI are realm names for the Product Catalog and Single API. Activities such as publishing policies and resynchronizing policies rely on this naming convention.

When creating an authorization policy, you must provide a *policy name* to uniquely identify the policy, the security *realm name* that the policy applies to, and one or more *rules* for the policy. For multiple rules, separate them with commas and enclose the group of rules in quotation marks. It is recommended that you also provide a *policy description*. You can also optionally provide a *rule-combining algorithm*. Valid values for the algorithm are permit-overrides, deny-overrides, or first-applicable (permit-overrides means that if any rule evaluates to `Permit`, the final authorization decision is also `Permit`; deny-overrides means that if any rule evaluates to `Deny`, the final authorization decision is also `Deny`; first-applicable means that the result of the first relevant rule encountered is the final authorization decision). If you do not provide a rule-combining algorithm, the default is permit-overrides.

The example in [Figure 26](#) shows how to create an authorization policy using the Management Shell CLI. In this example, no rule-combining algorithm is specified, so the algorithm is permit-overrides, the default algorithm.

Figure 26 CLI create_auth_policy Example

```
upm1:root:mshell> create_auth_policy -id DEMO_DEFAULT_ACCESS -description  
"Demo Default Access" -realm DEMO_REALM1 -rules "PERMIT_ALL_DEMO,DENY_ALL"  
Status Message:  
    Successfully added policy DEMO_DEFAULT_ACCESS  
upm1:root:mshell>
```



NOTE

After a policy is created, it must be published before it can be used by an application. See [“Publishing Policies” on page 43](#).

Modifying Policies

After an authorization policy has been created, you can modify the policy description, the security realm the policy is associated with, the rule-combining algorithm, and the list of rules associated with the policy. For the rule-combining algorithm, valid values are permit-overrides, deny-overrides, and first-applicable (permit-overrides means that if any rule evaluates to `Permit`, the final authorization decision is also `Permit`; deny-overrides means that if any rule evaluates to `Deny`, the final authorization decision is also `Deny`; first-applicable means that the result of the first relevant rule encountered is the final authorization decision).

When modifying an authorization policy, you must provide the *policy name* and the information you want to modify. The example in [Figure 27 on page 43](#) shows how to modify an authorization policy using the Management Shell CLI. This example illustrates changing the list of rules associated with the policy.

Figure 27 CLI modify_auth_policy Example

```
upm1:root:mshell> modify_auth_policy -id DEMO_DEFAULT_ACCESS -rules "PERMIT_ADMIN,DENY_ALL"
Status Message:
    Successfully updated policy DEMO_DEFAULT_ACCESS
upm1:root:mshell>
```

**NOTE**

Modifications to a policy only affect the database representation of the policy and do not affect the currently published policy being used by an application. The policy must be republished before changes take effect for the target application. See [“Publishing Policies” on page 43](#). Republishing a policy is necessary only for such modifications as changes to rules or the rule-combining algorithm. A change to the policy description does not require republishing.

Deleting Policies

You can delete an individual authorization policy or a group of policies with a common prefix in their names. (For example, if multiple policies are named *ABC<SOMETHING>*, *ABC* is the prefix.) When deleting authorization policies, you must provide the unique *policy name* or the *prefix* in the names. If you provide a prefix, all policies with a matching prefix are deleted. The example in [Figure 28](#) shows how to delete an individual authorization policy using the Management Shell CLI.

Figure 28 CLI remove_auth_policy Example

```
upm1:root:mshell> remove_auth_policy -id DEMO_DEFAULT_ACCESS
Status Message:
    Successfully deleted policy DEMO_DEFAULT_ACCESS
upm1:root:mshell>
```

Deleting a policy removes it from the Security Server database. If the policy has been published, you must publish and resynchronize policies for the target node(s) again to remove the policy from the target node(s). See [“Publishing Policies”](#) below.

Publishing Policies

**NOTE**

Activities such as publishing policies and resynchronizing policies rely on a policy-naming convention. The first few characters of the policy name must be a realm name that identifies the realm to which the policy applies. Two examples of policy names are *PC_<SomeName>* and *SAPI_<SomeName>*, where *PC* and *SAPI* are realm names for the Product Catalog and Single API.

After an authorization policy is created using the `create_auth_policy` command, it exists in the Security Server database. However, a policy must be published, retrieved by the Unified

Platform Agent (UPA) residing on the same physical node as that of the target application, and loaded by the UPA before the application can use it. Similarly, if a currently published policy has been modified, or if any of its rules have been modified, the policy must be published (that is, republished) and resynchronized with the appropriate node(s) before these modifications take effect for the target application.

Publishing a policy means that the policy is exported from the database to XACML format (that is, to an XML file that conforms to the XACML schema). The exported XACML-formatted file is placed in the `$JBOSS_HOME/conf/policy` directory. When a UPA is restarted, it automatically resynchronizes policies by retrieving the appropriate policies from this directory and loading them for use. When publishing policies, you can specify the node(s) for which policies need to be resynchronized (such as SAPI, SDP, and so on). In this case, the UPA(s) on the appropriate node(s) retrieve and load the XACML-formatted policies on demand, without requiring a restart of the UPA. After a UPA loads a policy, authorization requests use the newly published policy.

You can publish all authorization policies defined in the Security Server database, an individual policy, or multiple policies with a common prefix in their names. When publishing an individual policy, you must provide the unique *policy name*. When publishing policies with a common prefix in policy names, you must provide the *prefix*. To optionally identify the target node(s) for resynchronizing the policies, you can provide a node *class* (such as SAPI, SDP, and so on), a *managed object name* (that is, a node name), or a node *instance* (that is, the IP address for the instance). The example in [Figure 29](#) shows how to publish an individual policy using the Management Shell CLI. This example illustrates only publishing the policy, without node resynchronization.

Figure 29 CLI publish_policy Example

```
upm1:root:mshell> publish_policy -id UP_DEFAULT_ACCESS
Publish Policy Information
```

Id	NodeClass	NodeName	NodeInstance	Status
UP_DEFAULT_ACCESS	MANAGER	--	10.230.12.57	Publish Success

```
upm1:root:mshell>
```

Resynchronizing Policies



NOTE

Activities such as publishing policies and resynchronizing policies rely on a policy-naming convention. The first few characters of the policy name must be a realm name that identifies the realm to which the policy applies. Two examples of policy names are `PC_<SomeName>` and `SAPI_<SomeName>`, where PC and SAPI are realm names for the Product Catalog and Single API.

If policies have been published (that is, exported to XACML-formatted files on the Security Server) but not retrieved by the Unified Platform Agents for the appropriate nodes, or if XACML-formatted policies have been created manually, you must resynchronize policies before they can be used by target applications.

You can resynchronize all XACML-formatted policies that reside on the Security Server, or you can optionally specify selected target nodes whose policies you want to resynchronize. To identify the target node(s) for the policies, you can provide a node *class* (such as SAPI, SDP, and so on), a *managed object name* (that is, a node name), or a node *instance* (that is, the IP address for the

instance). The example in [Figure 30](#) shows how to resynchronize policies using the Management Shell CLI. This example illustrates resynchronizing all policies for the SAPI node class.

Figure 30 CLI resync_policy Example

```
upm1:root:mshell> resync_policy -c sapi
```

Bulk Policy Operations

Instead of using the CLI's prompt-based mode to create authorization rules and policies, you can use a batch process for a bulk load operation. When many policies, or many rules, need to be created, bulk load is the recommended approach.

The input file for the batch process is a Microsoft Excel worksheet (also called a spreadsheet). A sample worksheet template, named `PolicyAdministrationTemplate.xls`, is installed during Security Server installation and is located in `$JBOSS_HOME/templates/security`.



NOTE

For bulk load operations, the 2007 Office Open XML format is not supported. This means that spreadsheets must be saved in the 2003 format (that is, with the `.xls` extension instead of the `.xlsx` extension).

For the bulk load operation, do the following five steps:

1. Using the worksheet template as a guide, create a new worksheet file.
2. Complete the worksheet, following the instructions and comments provided in the template.
3. Save your worksheet file with a filename that conforms to the appropriate format. The format for the filename is `<NodeClass>_<SecurityRealm>_<PolicyTag>.xls`, where `<NodeClass>` is the node class to which the policy applies (such as SAPI, UP, SDP, and so on), `<SecurityRealm>` is the security realm that the policy is associated with and `<PolicyTag>` is a short descriptive term that provides some context.
4. Place the completed worksheet on the Security Server in either the default location (`$JBOSS_HOME/batch/policy` directory) or in a directory of your choice.
5. To process policy worksheets for the bulk load operation, do one of the following steps:
 - a. If the worksheet is located in the default location, execute the following command using the Management Shell CLI:
create_auth_policy -b
 This command processes policy worksheets in the `$JBOSS_HOME/batch/policy` directory and loads the data into the database.
 - b. If the worksheet is located in another directory, specify the directory in the command. For example, if the directory is `/tmp`, execute the following command using the Management Shell CLI:
create_auth_policy -b /tmp

This command processes policy worksheets in the `/tmp` directory and loads the data into the database.

Worksheet files remain in the directory where they were placed unless you manually remove them.

As previously mentioned, when authorization policies are initially created, they exist only in the database. To activate the policies for use by applications, you must publish them. See [“Publishing Policies” on page 43](#).

Manual Policy Creation

As stated earlier in this chapter, the Comverse security solution supports the full XACML specification, but the interface that allows for policy creation has been simplified to expose only the XACML properties that are required for Comverse ONE components. The Security Server database model does not support every option provided in the XACML schema. If more advanced XACML policies are required, they must be defined outside the Security Server interface in the XACML native format.

To create a more advanced XACML policy, do the following three steps:

1. Manually create a policy file that conforms to the XACML XML schema definition (XSD). The schema is located here: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-policy-schema-os.xsd.
2. Name the policy file with a filename that conforms to the following naming convention:
`<NodeClass>_<SecurityRealm>_<PolicyTag>.xml`
where `<NodeClass>` is the node class to which the policy applies (such as SAPI, UP, SDP, and so on), `<SecurityRealm>` is the security realm that the policy is associated with, and `<PolicyTag>` is a short descriptive term that provides some context.
3. Place the policy file on the Security Server in the `$JBOSS_HOME/conf/policy` directory.

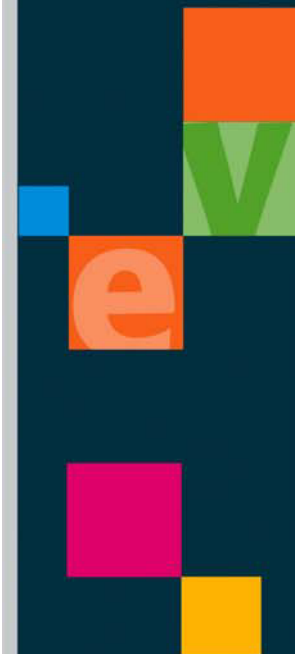
Before a manually created policy can be used by an application, it must be retrieved by the Unified Platform Agent (UPA) residing on the same physical node as that of the application and loaded by the UPA. Use the `resync_policy` command for this purpose. For details, see [“Resynchronizing Policies” on page 44](#).

Chapter 5

Audit Management

5

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To



Audit Management Overview

Audit management provides complex systems the ability to centrally log data in a consistent format for the purpose of post-incident analysis in the event of application system security breaches or for root cause analysis (RCA).

Comverse's audit management solution is primarily focused on SARBOX security audit information for the purpose of your organization's security audits. This means that Comverse ONE applications mainly audit activities that directly or indirectly affect the financial data and reporting of the system.

For audit management purposes, the Comverse ONE solution uses the audit record format, audit event/event outcome codes, and client API defined in the Distributed Audit Service (XDAS) specification. The governing body for XDAS is The Open Group, which is a vendor-neutral and technology-neutral consortium whose stated mission is to enable access to integrated information, within and between enterprises, based on open standards and global interoperability. Information is available from the following website: www.opengroup.org.

In the Comverse ONE solution, the Unified Platform Agent (UPA) on managed nodes handles the periodic forwarding of audit records to the Security Server. If the UPA, Security Server, or both are down, audit records from the local application such as the Unified API (also known as the Single API, or SAPI) are persisted to `$JBOSS_HOME/sdkaudit` in encrypted form. Once the UPA and Security Server are restored, the persisted audit records are processed, forwarded to the Security Server, and purged from the local filesystem. This approach helps ensure that audit records are not lost during UPA/Security Server downtime.

Getting Started with Audit Management

Topics in this chapter include audit operations tasks, the audit record format, and the audit event index (audit event codes and event outcome codes).

Audit Management Interface

The Security Server provides a command line interface (CLI) used to perform audit management tasks and a graphical user interface (GUI) that permits viewing the list of audit records. This chapter discusses the CLI. (For information on the GUI, see [Chapter 7, "Security GUI."](#)) The two modes of operation for the CLI are interactive mode and noninteractive mode as explained below:

- Interactive mode provides a prompt-based CLI. Information in this chapter deals with CLI commands, accessed via the Management Shell (mshell), in interactive prompt-based mode. To run the Management Shell, log in as the `root` user (using an SSH2 connection if you are accessing remotely), type **mshell** at the command line, and provide your security administrator username/password when prompted.
- Noninteractive mode enables you to automate tasks using a noninteractive scriptable interface.

In noninteractive mode, you can execute the following command at the command line to run the Management Shell, log in, and run the specified command:

```
mshell <username>/<password> <command>
```

The scenario above is what allows for scripts to use noninteractive mode.

As a second scenario for noninteractive mode, you can create a text file of valid CLI commands, with one command on each line. Place the file in a location of your choice. Execute the following command at the command line to run the Management Shell, log in, and run commands in the file:

```
mshell <username>/<password> @<filename>
```

where @<filename> is an absolute path to the file (for example, /tmp/commands.txt)
This second scenario is not meant for scripts.

Audit Management Tasks

To go directly to a particular task, use the references below:

- [“Enabling/Disabling Auditing” on page 50](#)
- [“Querying Audit Records” on page 50](#)
- [“Purging Audit Records” on page 51](#)

Audit Management Operations

The following sections discuss audit management operations.

Enabling/Disabling Auditing

In the Comverse ONE solution, each application has its own method to enable or disable auditing. In future releases, a central control mechanism will be provided to enable/disable auditing for applications.



NOTE

Be aware that disabling auditing in a production environment compromises SARBOX compliance.

Querying Audit Records

To query audit records, use the `build_report` command. This command is used for many other purposes (see the *Unified Platform Guide*), but information in this section is specific to audit records.

When using the `build_report` command to query audit records and view results in a report, you must provide the *report type* (audit). You can optionally provide a *beginning date* and *ending date* to specify the time period for the report. If you do not provide beginning and ending dates, by default the report includes audit records for the last hour. (If a beginning date and ending date are provided, the difference between these dates cannot be more than one day, such as “11/03/2008” for the beginning date and “11/04/2008” for the ending date.) You can also optionally provide a *user ID* to limit the report to only audit records for a specific user, a *command name* to limit the report to only audit records corresponding to a specific command (for example, `login`), or an *external ID* (such as a subscriber number) to limit the report to only audit records corresponding to that ID. The example in [Figure 31](#) shows how to query audit records using the Management Shell CLI.

Figure 31 CLI `build_report` Example

```
upml:root:mshell> build_report -r audit -b "11/06/2008" -e "11/07/2008" -cn login
```

Time Offset	User Name	Event Outcome	Event Number	Originator Address	Originator Name	Target PrincipalName	Event Info
2008-11-06 01:28:42.0	secadmin	0	16777223	10.210.156.164	devsite/upsec/upml/manager	Login	command=Login,method=null,externalid=--
2008-11-06 09:27:06.0	secadmin	0	16777223	10.210.156.164	devsite/upsec/upml/manager	Login	command=Login,method=null,externalid=--
2008-11-06 09:41:29.0	pcuser	0	16777223	10.210.156.164	devsite/upsec/upml/manager	Login	--,externalid=--
2008-11-06 10:48:43.0	secadmin	0	16777223	10.210.156.164	devsite/upsec/upml/manager	Login	command=Login,method=null,externalid=--

```
upml:root:mshell>
```

For details on information included in the audit report, see [“Audit Record Format” on page 51](#). In the Event Outcome and Event Number columns, the report shows decimal representations of the

hex XDAS codes discussed in [“XDAS Event Outcome Codes” on page 57](#) and [“XDAS Event Codes” on page 53](#).



NOTE

The XDAS specification states that the hexadecimal codes should be displayed as decimal.

Purging Audit Records

The amount of audit record data that is maintained depends on database sizing and capacity and your organization’s requirements. Audit record purging is handled by an automated job called `purge_audit` that purges old audit records.

For more details on jobs and how to execute them, see the *Unified Platform Guide*.

Audit Record Format

In the Comverse ONE solution, the audit record format conforms to the XDAS audit record format. The audit record is a UTF-8, length-preceded string that contains colon-delimited text fields. Audit records are divided into the following six sections:

- Header
- Originator
- Initiator
- Target
- Source
- Event-specific data

The following represents the audit record format:

```
HDR:<hex_length_in_bytes>:<time_offset>:<event_number>:<hex_outcome>:
ORG:<loc_name>:<loc_address>:<service_type>:<authentication_authority>:
INT:<authentication_authority>:<domain-specific name>:<domain-specific id>:
TGT:<loc_name>:<loc_addr>:<service_type>:<authentication_authority>:
SRC:<external_source>:
EVT:<name1>=<value1>,<name2>=<value2>:
END:<end_of_record>
```

The layout above appears on multiple lines for the sake of presentation. Text within angle brackets represents field names. With the exception of line breaks, all other text is literal. In reality, no embedded line breaks exist between fields. Fields are delimited only by the intervening colon characters.

Each field is required, but some are automatically generated by the Security Server API. For example, the HDR:, ORG:, INT:, TGT:, SRC:, EVT:, and END: tags are never part of any data submitted through the Security Server API interfaces. Additionally, as discussed below, most fields in the header section are generated fields.

Header

Header information is common to all events. This data includes the following fields:

- **Record Length:** A four-digit hexadecimal number representing the length of the record.

- **Time Offset:** A timestamp field that contains eight hexadecimal digits whose value represents the number of seconds since the beginning of the epoch (midnight on January 1, 1970).
- **Event Number:** (Provided by the client application creating the audit record.) A set of semantic meanings for events that are likely to meet most application auditing needs. By providing a standardized generic event taxonomy, both application developers and analysis tool writers can easily understand the intended meaning of a given generic event. See [“XDAS Event Codes” on page 53](#).
- **Event Outcome:** (Provided by the client application creating the audit record.) Represents a standardized event outcome code as defined in the XDAS specification. See [“XDAS Event Outcome Codes” on page 57](#).

Originator

The originator is the entity that is detecting the auditable event and requesting the recording of the event. Within the originator information, the following data elements are specified:

- **Location Name:** Name of the originator host or service that is reporting the event. This name is an XDAS composite name, which is composed of a hierarchical representation of services.
- **Location Address:** A communication service end-point address, which can be a URL or other type of address that fully specifies a connection point.
- **Service Type:** Indicates the protocol used by the Location Address.
- **Authentication Authority:** Name of a server, or domain and realm, that provides the identities involved in the associated events. A UNIX hostname is a good example of an Authentication Authority. The Security Server is another example. This name field follows the same formatting rules as the Location Name field.
- **Principal Name:** User name relative to the Authentication Authority. For example, if the application is the Apache Web server, the originator Principal Name might be the name of the UNIX user account under which Apache is running. The Principal Name is optional and can be left blank.
- **Principal ID:** User ID of the principal, relative to the Authentication Authority.

Initiator

The initiator is the user or identity that causes the auditable event. For example, a user logging in through a Web server is the initiator. The following fields are associated with the initiator string and are specified in the following order, separated by colon characters:

- **Authentication Authority:** Host, service, or domain and realm name, of the authentication service that provides the identity attributes for the initiator of an auditable action. An example is the Security Server. Another example is the client organization’s authentication authority if the Security Server is not being used for authentication.
- **Domain-Specific Name:** Name of the user or identity that is initiating an auditable action, relative to the initiator Authentication Authority.
- **Domain-Specific ID:** Identifier (user ID, globally unique ID, and so on) of the user or identity that is initiating the auditable action, relative to the initiator Authentication Authority.

Target

The target is the object being acted on during the auditable event. For example, a user whose rights are being modified through a Web management interface is a target. These fields are similar

in structure and content to the originator fields defined previously, except these fields indicate information about the object being acted on, not the actor.

**NOTE**

Not all events have targets with associated identities. If that is the case, the last three fields in the following list are optional and are left blank.

The following fields are associated with the target information fields of an XDAS record, defined in the following order:

- **Location Name:** Name of the target resource, host, or service of the event that is being reported. This name is an XDAS composite name, which is composed of a hierarchical representation of services or resources.
- **Location Address:** Communication service end-point address, which can be a URL or other type of address that fully specifies a connection point.
- **Service Type:** Indicates the protocol used by the target Location Address.
- **Authentication Authority** (optional): Name of a server, or a domain and realm, that provides identities involved in the associated events. The target authentication information might be left blank in this case.
- **Principal Name** (optional): User name relative to the target Authentication Authority.
- **Principal ID** (optional): User ID of the target principal, relative to the Authentication Authority.

Source

The source field contains a pointer to a source domain in case the event was imported from a domain-specific logging service. This allows the audit record to contain only security-relevant information without losing the ability to reference the additional information logged with the original event.

**NOTE**

The Source field is not currently used.

Event

The event data field is designed to carry additional information that is specific to the application domain. The XDAS specification indicates that this field is for environments where XDAS is the primary audit system, and events should carry domain-specific information as well as XDAS generic security event information. The format of this string is text-only, comma-delimited `<name>=<value>` pairs.

Audit Event Index

The following sections provide details about the XDAS event codes and event outcome codes used in audit records.

XDAS Event Codes

This section describes the XDAS event codes, separated into the following sets of events:

- Account management events
- User session events
- Data-item and resource-element management events
- Service or application management events
- Service or application utilization events
- Peer-association management events
- Data-item or resource-element content access events
- Exceptional events
- Audit service management events

Account Management Events

Accounts exist in application domains in order to persistently associate attributes with the set of identifiers typically associated with identities. An identity, in this context, is a token used to represent a particular user or entity responsible for a set of activities within a system. This is not necessarily a human being, but instead can be an automated identity, such as another service that might be acting on behalf of a human or a regularly scheduled system activity. In any case, account management means any form of persistent account creation, wherein an identity is associated with attributes. The following is a list of XDAS event codes for account management events:

- **XDAS_AE_CREATE_ACCOUNT [0x01000001]:** This event is reported for any situation in which an account is created.
- **XDAS_AE_DELETE_ACCOUNT [0x01000002]:** This event has the opposite semantic meaning of account creation, and is reported wherever such an account (as described above) is deleted.
- **XDAS_AE_DISABLE_ACCOUNT [0x01000003]:** This event is reported for any situation where a particular record in an identity repository (such as LDAP) is disabled (by an administrator or an automated security process) such that it can no longer be used until it is re-enabled.
- **XDAS_AE_ENABLE_ACCOUNT [0x01000004]:** This is the counterpart event to the disable account event defined above.
- **XDAS_AE_QUERY_ACCOUNT [0x01000005]:** Query account events are reported whenever a request for attribute information for a particular account is made.
- **XDAS_AE_MODIFY_ACCOUNT [0x01000006]:** Modify account events are reported whenever a request to change attribute information for a particular account is made.

User Session Events

The abstract concept of a session can be explained as the association of an initiator with a stream of communication. A session might represent a user's connection to a server, or a set of related transactions in a connectionless environment. The following is a list of XDAS event codes for user session events:

- **XDAS_AE_CREATE_SESSION [0x01000007]:** This event is reported whenever a new session (as defined above) is created.
- **XDAS_AE_TERMINATE_SESSION [0x01000008]:** This event is reported whenever an existing session (as defined above) is terminated.
- **XDAS_AE_QUERY_SESSION [0x01000009]:** This event is reported whenever attribute information is requested on an existing session.
- **XDAS_AE_MODIFY_SESSION [0x0100000A]:** This event is reported whenever attribute information is modified on an existing session.

Data-Item and Resource-Element Management Events

This set of events relates to the creation and management of data items and resource elements within a domain. The type of data item or resource element depends on the domain. For example, resources can be files and directories, tables and records within a database, or application-specific messages. The term “data item” in this context refers to any type of resource element. The following is a list of XDAS event codes for data-item and resource-element management events:

- **XDAS_AE_CREATE_DATA_ITEM [0x0100000B]**: This event is reported whenever a security-relevant data item is created.
- **XDAS_AE_DELETE_DATA_ITEM [0x0100000C]**: This event is reported whenever a security-relevant data item is deleted.
- **XDAS_AE_QUERY_DATA_ITEM_ATT [0x0100000D]**: This event is reported whenever a security-relevant data item is queried, either the value or an attribute of the data item.
- **XDAS_AE_MODIFY_DATA_ITEM_ATT [0x0100000E]**: This event is reported whenever a security-relevant data item is modified, either the value or an attribute of the data item.

Service or Application Management Events

This set of events relates to the management of services or applications. For example, the Red Hat Package Manager (RPM) reports these events as packages are installed or removed from a UNIX system. This set of events can also be much more domain-specific, including concepts such as installing, removing, or configuring installable executable modules within a single application domain. The following is a list of XDAS event codes for service or application management events:

- **XDAS_AE_INSTALL_SERVICE [0x0100000F]**: This event is reported when a service or application has been installed.
- **XDAS_AE_REMOVE_SERVICE [0x01000010]**: This event is reported when a service or application has been removed.
- **XDAS_AE_QUERY_SERVICE_CONFIG [0x01000011]**: This event is reported when service or application configuration information is requested.
- **XDAS_AE_DISABLE_SERVICE [0x01000013]**: This event is reported when service or application configuration information is disabled.
- **XDAS_AE_ENABLE_SERVICE [0x01000014]**: This event is reported when a service, operation, or function is enabled.

Service or Application Utilization Events

This class of events relates to the use of services and applications. The events typically map to the execution of a program or a procedure and manipulation of the processing environment. The following is a list of XDAS event codes for service or application utilization events:

- **XDAS_AE_INVOKE_SERVICE [0x01000015]**: This event is reported when a security-relevant service is invoked.
- **XDAS_AE_TERMINATE_SERVICE [0x01000016]**: This event is reported when a security-relevant service is terminated.
- **XDAS_AE_QUERY_PROCESS_CONTEXT [0x01000017]**: This event is reported when any attributes of a process context are queried. This event is somewhat specific to operating systems, but it also might be reported in other domain-specific applications.
- **XDAS_AE_MODIFY_PROCESS_CONTEXT [0x01000018]**: This event is reported when any attributes of a process context are modified. This event is somewhat specific to operating systems, but it also might be reported in other domain-specific applications.

Peer-Association Management Events

Peer-association events are related to the association of a user or identity with a group, or the association of two users in some domain-specific context. An example might be adding an LDAP user to a group, or associating two users for a domain-specific purpose in an application's identity management database. The following is a list of XDAS event codes for peer-association management events:

These events can also be used to associate identities within disparate authentication domains for purposes of federation. For example, when an identity in domain A makes a request to a service in domain B, then a peer association is required between these domains. This association is often known as a trust relationship.

- **XDAS_AE_CREATE_PEER_ASSOC [0x01000019]:** This event is reported when a new peer association is created.
- **XDAS_AE_TERMINATE_PEER_ASSOC [0x0100001A]:** This event is reported when an existing peer association is destroyed.
- **XDAS_AE_QUERY_ASSOC_CONTEXT [0x0100001B]:** This event is reported when the attributes of a peer association are queried.
- **XDAS_AE_MODIFY_ASSOC_CONTEXT [0x0100001C]:** This event is reported when the attributes of a peer association are modified.
- **XDAS_AE_RECEIVE_DATA_VIA_ASSOC [0x0100001D]:** This event is reported when data is received from a service in an authentication domain specifically via a trust relationship or peer association.
- **XDAS_AE_SEND_DATA_VIA_ASSOC [0x0100001E]:** This event is reported when data is sent to a service in an authentication domain specifically via a trust relationship or peer association.

Data-Item or Resource-Element Content Access Events

Resource content access events are related to access of any data files protected by an authentication domain. This can be file system files, database records, Web pages, and so on. Resource access can be a high-bandwidth process, so only security-relevant events are reported. The following is a list of XDAS event codes for data-item or resource-element content access events:

- **XDAS_AE_CREATE_DATA_ITEM_ASSOC [0x0100001F]:** This event is reported when rights are granted to an identity to a specific data item (that is, when a trust relationship is established between an identity and a data item).
- **XDAS_AE_TERMINATE_DATA_ITEM_ASSOC [0x01000020]:** This event is reported when rights are revoked from an identity to a specific data item (that is, when a trust relationship is revoked between an identity and a data item).
- **XDAS_AE_QUERY_DATA_ITEM_ASSOC_CONTEXT [0x01000021]:** This event is reported when rights are queried for an identity and a specific data item (that is, when trust relationship attributes are queried for a specific identity and data item).
- **XDAS_AE_MODIFY_DATA_ITEM_ASSOC_CONTEXT [0x01000022]:** This event is reported when rights are modified on the previously established relationship between an identity and specific data item.
- **XDAS_AE_QUERY_DATA_ITEM_CONTENTS [0x01000023]:** This event is reported when a data item is read on behalf of an identity.
- **XDAS_AE_MODIFY_DATA_ITEM_CONTENTS [0x01000024]:** This event is reported when a data item is written on behalf of an identity.

Exceptional Events

Exceptional events are events that do not happen often and that are considered important simply because they happened. For instance, shutting down an enterprise-critical server is exceptional because it should not happen without someone's permission. The following is a list of XDAS event codes for exceptional events:

- **XDAS_AE_START_SYS [0x01000025]:** This event is reported when a server, system, or mission-critical application starts up.
- **XDAS_AE_SHUTDOWN_SYS [0x01000026]:** This event is reported when a server, system, or mission-critical application shuts down.
- **XDAS_AE_RESOURCE_EXHAUST [0x01000027]:** This event is reported when a server, system, or mission-critical application runs out of some critical resource (memory, disk space, and so on). Note that it is often difficult to report such events because often the critical resource in question is required in order to report the event.
- **XDAS_AE_RESOURCE_CORRUPT [0x01000028]:** This event is reported when a server, system, or mission-critical application detects a resource corruption (memory, disk file, and so on).
- **XDAS_AE_BACKUP_DATASTORE [0x01000029]:** This event is reported when a server, system, or mission-critical application backs up a critical data store.
- **XDAS_AE_RECOVER_DATASTORE [0x0100002A]:** This event is reported when a server, system, or mission-critical application restores a critical data store.

Audit Service Management Events

For a variety of reasons, audit services have traditionally been classified by themselves. This is probably because auditing represents a lower-level activity than the security events themselves being audited. By classifying audit events separately, an entire category of endless loop defects can be avoided.

It is possible that applications will never report these events, as they are generally reported by the audit system itself. However, they are documented here for the sake of completeness. The following is a list of XDAS event codes for audit service management events:

- **XDAS_AE_AUD_CONFIG [0x0100002B]:** Configuration data has been changed for an audit subsystem. This event is reported when a command triggering the rereading of the configuration is executed (for example, a SIGHUP signal or reload command).
- **XDAS_AE_AUD_DS_FULL [0x0100002C]:** This event is reported when an audit log is full and can no longer accept additional audit records. Where possible, space is reserved for this event, in case it must be reported.

XDAS Event Outcome Codes

The generic event outcome codes are broken into the following four sets of codes:

- Nonclassified event outcome codes
- Success event outcome codes
- Failure event outcome codes
- Denial event outcome codes

The first set contains only the value 0xFFFFFFFF and represents the choice not to specify the outcome code at this time. This choice is necessary because the outcome of an event might not be known at the time the event record is initially created. The second set represents successful outcome codes. The third set represents failed outcome codes. The last set represents denial outcome codes.

Careful examination of the bit patterns of the values associated with each outcome type shows that the outcome code classes are defined by the bits of the least significant byte. The class defined by FF in this byte indicates “unspecified,” while 00 indicates success, 01 indicates failure, and 02 indicates denial.

Nonclassified Event Outcome Codes

- **XDAS_OUT_NOT_SPECIFIED [0xFFFFFFFF]:** This outcome code is not really an outcome code at all, but rather a special value passed when the client application does not specify an outcome code. All audit records must ultimately have a valid outcome code (other than this value) before they can be submitted.

Success Event Outcome Codes

- **XDAS_OUT_SUCCESS [0x00000000]:** This code indicates pure success, with no caveats or side-band qualifications. This code is used in the absence of a better, more specific success code.
- **XDAS_OUT_PRIV_USED [0x00000100]:** This success code indicates that the requested privilege was successfully used in the operation.
- **XDAS_OUT_PRIV_GRANTED [0x00000200]:** This success code indicates that the requested privilege was successfully granted.
- **XDAS_OUT_PRIV_REVOKED [0x00000400]:** This success code indicates that the requested privilege was successfully revoked.
- **XDAS_OUT_PRESELECT_CRITERIA_SET [0x00000800]:** This success code indicates that the requested preselection criterion was successfully set.
- **XDAS_OUT_THRESHOLDS_SET [0x00001000]:** This success code indicates that the requested thresholds were successfully set.
- **XDAS_OUT_ACTIONS_SET [0x00002000]:** This success code indicates that the requested actions were successfully set.

Failure Event Outcome Codes

- **XDAS_OUT_FAILURE [0x00000001]:** This code indicates pure failure, with no caveats or side-band qualifications. This code is used in the absence of a better, more specific failure code.
- **XDAS_OUT_SERVICE_UNAVAILABLE [0x00000101]:** This failure code indicates that the specified service was unavailable.
- **XDAS_OUT_SERVICE_FAILURE [0x00000201]:** This failure code indicates that the specified service failed to successfully complete the requested operation.
- **XDAS_OUT_HARDWARE_FAILURE [0x00000401]:** This failure code indicates a hardware failure of some sort, directly causing the requested operation to fail.
- **XDAS_OUT_LOST_ASSOCIATION [0x00000801]:** This failure code indicates that the request could not be completed due to a lost association. More specifically, this outcome code is probably means that a trusted association between identities on disparate systems is no longer valid, and thus the operation could not be completed due to security or rights issues.
- **XDAS_OUT_ALREADY_ENABLED [0x00001001]:** This failure code indicates that the request to enable a service, operation, or function failed because the target is already enabled.
- **XDAS_OUT_ALREADY_DISABLED [0x00002001]:** This failure code indicates that the request to disable a service, operation, or function failed because the target is already disabled.

- **XDAS_OUT_SERVICE_ERROR [0x00004001]:** This failure code indicates that the requested operation failed because a primary or secondary service, operation, or function failed an intermediate or direct request associated with the audited operation.
- **XDAS_OUT_BUSY [0x00008001]:** This failure code indicates that the requested operation failed because a required intermediate or end-point service was busy.
- **XDAS_OUT_DISABLED [0x00010001]:** This failure code indicates that the requested operation failed because a required intermediate or end-point service was disabled.
- **XDAS_OUT_INVALID_INPUT [0x00020001]:** This failure code indicates that the requested operation failed because some parameter or input to an intermediate or end-point service was not valid according to that service.
- **XDAS_OUT_ENTITY_EXISTS [0x00040001]:** This failure code indicates that a request to create an entity failed because such an entity already exists.
- **XDAS_OUT_ENTITY_NON_EXISTENT [0x00080001]:** This failure code indicates that the request to query, delete, or otherwise access a resource failed because the target entity does not exist.

Denial Event Outcome Codes

- **XDAS_OUT_INSUFFICIENT_PRIVILEGE [0x00000102]:** This denial code indicates that the requested operation failed because of insufficient privileges. This is the semantic equivalent of the more widely understood “access denied” error.
- **XDAS_OUT_INVALID_IDENTITY [0x00000202]:** This denial code indicates that a target identity is invalid. Note that this denial code does *not* indicate anything about the initiating identity. As such, a response to a request could be a security risk (giving too much information about why the operation failed). In the case of an invalid initiator identity, XDAS_OUT_INSUFFICIENT_PRIVILEGE is used instead.
- **XDAS_OUT_INVALID_CREDENTIALS [0x00000402]:** This denial code indicates that a set of credentials presented during the operation was invalid. The set of credentials might be an intermediate set or a primary set.

6

Chapter 6

Encryption Key and Credentials Management

orded (sender
The desTir
noTiEying
ng The noTi
ieve The m
cT access To

e

v

Encryption Key and Credentials Management Overview

The Comverse security solution (1) provides client applications with an API to perform data encryption and (2) provides centralized management of symmetric encryption keys via the Security Server. The API for data encryption provides for both symmetric key encryption and asymmetric key (public key/private key) encryption. (An API is also available for digital signatures and message digests.) This chapter deals with the centrally managed symmetric encryption keys.

Symmetric key encryption uses the same key to both encrypt and decrypt data. In the Comverse ONE solution, a symmetric key can be created in one of two ways: (1) the client application requests creation of the key by means of the API, or (2) the security administrator creates the key manually.

When the client application requests creation of a key, the key management server, which is hosted on the Security Server, generates the key, assigns it a unique global key ID (GKID), stores the key in the Security Server database, and returns the key to a local keystore for use by the application.

When the security administrator creates a key manually, which also stores the key in the Security Server database, a custom name can be specified for the key ID. An application using symmetric key encryption either has a well-known key ID (custom name) that it uses to look up the key, or is configured with the GKID that was created when creating the key. Because a symmetric key is used for decryption as well as encryption, it must be kept secure. Centrally storing symmetric keys in the Security Server database overcomes the traditional key-management issues when using symmetric encryption algorithms

Converged only

An example of symmetric key encryption in the Comverse ONE solution is related to compliance with the Payment Card Industry Data Security Standard (PCI DSS). The Unified API (also called the Single API, or SAPI) uses a symmetric key to encrypt sensitive data fields (such as account number, cardholder name, expiration date) before storing that data in the Billing database.

For GUIs whose users have privileges to access sensitive data, such as the Credit Card Investigation Unit (CCIU), the symmetric key is used to decrypt the sensitive data fields. Also, back-end Billing and Financials payment modules, such as the Credit Card Payment Module (CPM), use a symmetric key to encrypt entire payment request files before writing to disk. The Unified Platform Agent running on the Billing database handles file transmission to clearinghouses and uses the key to decrypt the files before transmitting them over a secure channel. Similarly, incoming payment response files received from clearinghouses are encrypted before being written to disk and then decrypted for processing.

In addition to encryption key management, the security solution provides centralized credentials management. The term “credentials” in this context refers to (1) database credentials, which are passwords for business databases used by Comverse ONE applications, and (2) network credentials, which are SNMP community strings for network devices.

All database and network credentials are centrally managed at the Security Server. (In the future, other credentials such as operating system user passwords will be centrally managed. Currently, only database and network credentials are managed.) During initialization, applications retrieve the database credentials for one or more target databases. Centrally managing these credentials provides your organization the flexibility to change database passwords to conform to your own security policies. Changes made to database credentials from the Security Server are propagated

to the target databases. Network credentials are used by the Unified Platform to monitor unmanaged network devices.

Getting Started with Key/Credentials Management

Topics in this chapter include operations tasks for symmetric key management and credentials management (for database passwords and network-device SNMP community strings).

Key and Credentials Management Interface

The Security Server provides a command line interface (CLI) and a graphical user interface (GUI) used to perform symmetric key and credentials management tasks. This chapter discusses the CLI. (For information on the GUI, see [Chapter 7, “Security GUI.”](#)) The three modes of operation for the CLI are interactive mode, noninteractive mode, and batch mode as explained below:

- Interactive mode provides a prompt-based CLI. Information in this chapter deals with CLI commands, accessed via the Management Shell (mshell), in interactive prompt-based mode. To run the Management Shell, log in as the `root` user (using an SSH2 connection if you are accessing remotely), type `mshell` at the command line, and provide your security administrator username/password when prompted.
- Noninteractive mode enables you to automate tasks using a noninteractive scriptable interface.

In noninteractive mode, you can execute the following command at the command line to run the Management Shell, log in, and run the specified command:

```
mshell <username>/<password> <command>
```

The scenario above is what allows for scripts to use noninteractive mode.

As a second scenario for noninteractive mode, you can create a text file of valid CLI commands, with one command on each line. Place the file in a location of your choice. Execute the following command at the command line to run the Management Shell, log in, and run commands in the file:

```
mshell <username>/<password> @<filename>
```

where @<filename> is an absolute path to the file (for example, `/tmp/commands.txt`)

This second scenario is not meant for scripts.

- Batch mode is a special interactive mode that provides “bulk load” operations for credentials (that is, database passwords and/or SNMP community strings). See [“Bulk Credentials Operations” on page 70](#).

Key and Credentials Management Tasks

To go directly to a particular task, use the references below:

- [“Viewing Symmetric Keys” on page 65](#)
- [“Creating Symmetric Keys” on page 65](#)
- [“Disabling Symmetric Keys” on page 65](#)
- [“Enabling Symmetric Keys” on page 66](#)
- [“Deleting Symmetric Keys” on page 66](#)
- [“Viewing Credentials” on page 66](#)
- [“Creating Credentials” on page 67](#)
- [“Modifying Credentials” on page 68](#)
- [“Publishing Database Credentials” on page 69](#)

- [“Publishing Database Credentials” on page 69](#)

Symmetric Key Management Operations

The following sections discuss viewing, creating, changing, disabling, enabling, and deleting symmetric encryption keys.

Viewing Symmetric Keys

You can view all symmetric keys stored in the Security Server database. The example in [Figure 32](#) shows how to view keys using the Management Shell CLI.

Figure 32 CLI list_keys Example

```
upm1:root:mshell> list_keys
Symmetric Keys Listing.
```

GlobalKeyId	Status	Algorithm	CreationDate	KeyLength
PCI-COM	act	AES	2008-03-06 14:37:37.0	128
PCI_DB_FLD	act	AES	2008-03-06 21:55:04.0	128
01-16343162	act	Blowfish	2008-03-07 16:35:19.0	128
01-18708884	act	AES	2008-03-07 16:36:14.0	128

```
upm1:root:mshell>
```

Creating Symmetric Keys

When creating a symmetric key, you can optionally provide a unique *key ID* (custom name) and the encryption *algorithm* for which the key is being created. For example, you might need to provide a custom name when one or more components in the system are configured to retrieve a key with a well-known name and it is assumed that this name is provisioned as the key ID. The custom name can contain letters, digits, and special characters (such as underscore and hyphen). If you do not provide a key ID, the Security Server creates it. If the Security Server creates the key ID, the format is <SSID>-<KID>, where <SSID> is the Security Server ID and <KID> is a series of digits, with the combination resulting in a unique global key ID (for example, 01-16343162 or 02-18570884). Valid values for the encryption algorithm are AES and Blowfish. If you do not provide the algorithm, the default value is AES. The example in [Figure 33](#) shows how to create a symmetric key using the Management Shell CLI.

Figure 33 CLI create_key Example

```
upm1:root:mshell> create_key -kid 01-12345 -algo Blowfish
Status Message:
    Key created successfully.
upm1:root:mshell>
```

Disabling Symmetric Keys

Occasionally, you might need to disable a symmetric key manually. Disabling a symmetric key changes its status from active to inactive, meaning it is no longer usable. When disabling a symmetric key, you must provide the unique *key ID*. The example in [Figure 34 on page 66](#) shows how to disable a key using the Management Shell CLI.

Figure 34 CLI disable_key Example

```
upm1:root:mshell> disable_key -kid 01-12345
Status Message:
    Key disabled successfully
upm1:root:mshell>
```

Enabling Symmetric Keys

After a key has been disabled, it might be necessary to enable (activate) it again. When enabling a disabled symmetric key, you must provide the unique *key ID*. The example in [Figure 35](#) shows how to enable a key using the Management Shell CLI.

Figure 35 CLI enable_key Example

```
upm1:root:mshell> enable_key -kid 01-12345
Status Message:
    Key enabled successfully
upm1:root:mshell>
```

Deleting Symmetric Keys

In any situation where a key is no longer used, the old key should be deleted from the Security Server database. Before a symmetric key can be deleted, its status must be inactive. See [“Disabling Symmetric Keys”](#) above for information on changing the status to inactive. When deleting a symmetric key, you must provide the unique *key ID*. The example in [Figure 36](#) shows how to delete a key using the Management Shell CLI.

Figure 36 CLI delete_key Example

```
upm1:root:mshell> delete_key -kid 01-12345
Status Message:
    Key deleted successfully.
upm1:root:mshell>
```

Credentials Management Operations

The following sections discuss (1) viewing, creating, modifying, deleting, and publishing credentials in the Security Server database using the Management Shell CLI and (2) using a bulk load operation to create credentials. As mentioned previously, the term “credentials” in this context currently refers to (1) passwords for business databases used by Comverse ONE applications and (2) SNMP community strings for network devices.

Viewing Credentials

You can view credentials (database passwords or SNMP community strings for network devices) stored in the Security Server database. When viewing credentials, you can optionally provide a credential *type* (database or network) to limit the displayed credentials to only that specific type. If a type is not provided, the default is database. The example in [Figure 37 on page 67](#) shows how to

view credentials using the Management Shell CLI. In this example, no credential type is specified, so the displayed output is for database credentials only.

Figure 37 CLI list_credential Example

```
upm1:root:mshell> list_credential
DB Passwords Listing.

UserId                DBType      Instance    Password
-----
cbs_owner             RATING      MAIN        converse
upm1:root:mshell>
```

Creating Credentials

Currently, the credentials that can be created are database passwords for business databases used by Comverse ONE applications and SNMP community strings for network devices.

Creating Database Credentials

A database user (identified by a user ID, such as CBS_OWNER) must have the same password for a given instance across all databases of a given type in the system. For example, if RATING is the type and MAIN is the instance, and there are five RATING databases (that is, SDPs) in the system, then the password for the database user CBS_OWNER must be the same across all those five MAIN instances. For another instance of the RATING database (such as HISTORY), the password for user CBS_OWNER can be different from the password for the MAIN instance of the RATING database. (Use the `list_credential` command to see a listing of user IDs and passwords for all the database types/instances.)

If you specify `database` as the credential type when creating credentials, you must provide the unique database *user ID*, the database *instance name*, (such as MAIN or HIST if you provide a database type of RATING), and the *password* for the database user ID. In addition to the mandatory information, you can optionally provide the *database type*. Examples of some valid values for database type are RATING, PCAT, CBS, and so on. If you do not provide the database type, the default value is CBS. The example in [Figure 38](#) shows how to create a database password using the Management Shell CLI.

Figure 38 CLI store_credential Database Example

```
upm1:root:mshell> store_credential -type database -uid cbs_owner -dbtype RATING
-in MAIN -pwd converse
Status Message:
    DB password added successfully
upm1:root:mshell>
```



NOTE

After creation, a database password exists in the Security Server database. It must be propagated to the target database to activate it for use. For information, see [“Publishing Database Credentials” on page 69](#).

Creating Network Credentials

If you specify `network` as the credential type when creating credentials, you must provide the *node class*, *node name*, and *node instance* of the network device and the SNMP *community string* for the device. The example in [Figure 39](#) shows how to create credentials for a network device using the Management Shell CLI.

Figure 39 CLI `store_credential` Network Example

```
upm1:root:mshell> store_credential -type network -nc Cajun -nm Cajun1 -ni 0.20.0.0 -comm public
Status Message:
    Network node credential record added successfully
upm1:root:mshell>
```

Modifying Credentials

To modify a credential (database or network), you must first delete it (see [“Deleting Credentials”](#) below) and then create a new credential using the `store_credential` command.



NOTE

If a database password has already been propagated to the target database, deleting and recreating the password does not affect the password currently in use. For the change to take effect, you must propagate the new password to the target database. See [“Publishing Database Credentials”](#) on page 69.

Deleting Credentials

This section discusses deleting database credentials (that is, database passwords) and network credentials (that is, SNMP community strings for network devices).

Deleting Database Credentials

When deleting a database credential (database password), you must provide the credential *type* (database), the unique database *user ID*, and the database *instance*. In addition to the mandatory information, you can optionally provide the *database type*. If you do not provide a database type, a default value of CBS is assumed. The example in [Figure 40](#) shows how to delete a database password using the Management Shell CLI.

Figure 40 CLI `remove_credential` Database Example

```
upm1:root:mshell> remove_credential -type database -uid cbs_owner
-dbtype RATING -in MAIN
Status Message:
    DB password deleted successfully
upm1:root:mshell>
```

**NOTE**

Deleting a database credential (database password) deletes the information only from the Security Server database. If the password has been propagated to the target database, it remains in use until credentials are published.

Deleting Network Credentials

When deleting a network credential (SNMP community string for a network device), you must provide the credential *type* (network) and the *node class*, *node name*, and *node instance* of the network device. The example in [Figure 41](#) shows how to delete a network credential using the Management Shell CLI.

Figure 41 CLI remove_credential Network Example

```
upm1:root:mshell> remove_credential -type network -nc Cajun -nm Cajun1 -ni 0.20.0.0
Status Message:
    Network node credential record deleted successfully
upm1:root:mshell>
```

Publishing Database Credentials

**NOTE**

Publishing database credentials should be done during a maintenance window (that is, when the node where the database resides is in maintenance mode).

After database credentials (database passwords) are created, they must be published. Publishing credentials is a two-step process. First, execute the `publish_credential` command to propagate the password to the target database. Second, execute the `publish_credential` command again, specifying the refresh cache option, to update the password cache on the target node(s).

For the first execution of the command, you must provide the credential *type* (database is the only valid type), the unique database *user ID*, and the database *instance*. In addition to the mandatory information, you can optionally provide the *database type*. If you do not provide a database type, a default value of CBS is assumed. To identify the node(s) where the target database resides, you can provide a *node class* (that is, a node type such as SDP), a *managed object name* (that is, a node name), or a *node instance* (that is, the IP address for the instance).

For the second execution of the command, you must provide the same information as provided during the first execution and, in addition, specify the `-rc` option. The example in [Figure 42 on page 70](#) shows the two-step process of publishing a database password using the Management Shell CLI. This example illustrates publishing the password for a database that resides on nodes of the SDP class.

Figure 42 CLI publish_credential Example

```
upm1:root:mshell> publish_credential -type database -uid cbs_owner
-dbtype RATING -in MAIN -c sdp

upm1:root:mshell> publish_credential -type database -uid cbs_owner
-dbtype RATING -in MAIN -c sdp -rc
```

Bulk Credentials Operations

Instead of using the CLI's prompt-based mode to create credentials (database passwords or SNMP community strings), you can use a batch process for a bulk load operation.

The input file for the batch process is a Microsoft Excel worksheet (also called a spreadsheet). A sample worksheet template, named `CredentialManagement_Template.xls`, is installed during Security Server installation and is located in `$JBOSS_HOME/templates/security`.



NOTE

For bulk load operations, the 2007 Office Open XML format is not supported. This means that spreadsheets must be saved in the 2003 format (that is, with the `.xls` extension instead of the `.xlsx` extension).

For the bulk load operation, do the following five steps:

1. Using the worksheet template as a guide, create a new worksheet file.
2. Complete the worksheet, following the instructions and comments provided in the template.
3. Save your worksheet file with a descriptive filename. (The only requirement for the filename is that it have `.xls` as the extension.)
4. Place the completed worksheet on the Security Server in either the default location (`$JBOSS_HOME/batch/credentials` directory) or in a directory of your choice.
5. To process credentials worksheets for the bulk load operation, do one of the following steps:

- a. If the worksheet is located in the default location, execute the following command using the Management Shell CLI:

```
store_credential -b
```

This command processes worksheets in the `$JBOSS_HOME/batch/credentials` directory and loads the data into the Security Server database.

- b. If the worksheet is located in another directory, specify the directory in the command. For example, if the directory is `/tmp`, execute the following command using the Management Shell CLI:

```
store_credential -b /tmp
```

This command processes worksheets in the `/tmp` directory and loads the data into the Security Server database.

Worksheet files remain in the directory where they were placed unless you manually remove them.

As previously mentioned, when database credentials (database passwords) are initially created, they exist only in the Security Server database. To activate them for use, they must be propagated to the target databases. See [“Publishing Database Credentials” on page 69](#). Network credentials do not require publishing.

Operating System Credentials

Currently, operating system (OS) credentials are not centrally managed at the Security Server. To make changes to passwords for OS users, follow the standard UNIX guidelines using the `passwd` command.



CAUTION

Always record `root` user passwords along with the date. This information is essential in case the system has to be restored from an old OS backup copy.

Upgrades and Database/OS Credentials

During upgrades, all passwords for database users and OS users in the Comverse ONE solution will be reverted to the default passwords. After the upgrade, the passwords can be changed once again.

Changing Database Passwords after an Upgrade



NOTE

When changing database passwords, remember that a database user (identified by a user ID, such as `CBS_OWNER`) must have the same password for a given instance across all databases of a given type in the system. For example, if `RATING` is the type and `MAIN` is the instance, and there are five `RATING` databases (that is, `SDPs`) in the system, then the password for the database user `CBS_OWNER` must be the same across all those five `MAIN` instances. For another instance of the `RATING` database (such as `HIST`), the password for user `CBS_OWNER` can be different from the password for the `MAIN` instance of the `RATING` database.

To change default database passwords after an upgrade, do the following steps:

1. After logging in to `mshell`, use the `list_credential` command to get a listing of user IDs and passwords for all the database types/instances and make note of that information.
2. Delete the default passwords using the `remove_credential` command, discussed in [“Deleting Database Credentials” on page 68](#).
3. Create the new passwords using the `store_credential` command, discussed in [“Creating Database Credentials” on page 67](#).



NOTE

As an alternate to steps 2 and 3, you can use the [“Bulk Credentials Operations” on page 70](#). Follow instructions in the sample template to overwrite existing entries.

4. Publish the passwords using the `publish_credential` command, a two-step process discussed in [“Publishing Database Credentials” on page 69](#).

Changing OS User Passwords after an Upgrade

After an upgrade, the OS user passwords will be the default passwords. Follow the standard UNIX guidelines, using the `passwd` command, for changing these default passwords.



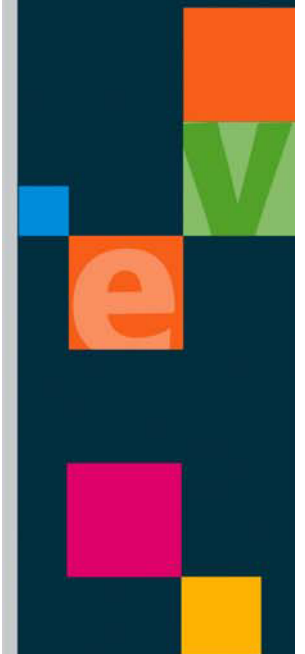
Always record `root` user passwords along with the date. This information is essential in case the system has to be restored from an old OS backup copy.

Chapter 7

Security GUI

7

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To



Security GUI

The Security graphical user interface (GUI) is a Web-based interface that is accessed via a standard Web browser. (Supported browsers are Internet Explorer 7 and 8 and Firefox 3.) The information in this chapter assumes you are familiar with concepts presented in previous chapters, so it focuses on GUI operations.



NOTE

This chapter provides details about all areas of the Security GUI. That is, it assumes you can view all areas and perform all actions in the GUI. A user's ability to view areas and perform actions in the GUI depends on the authorization rules defined in the Security GUI's authorization policy.

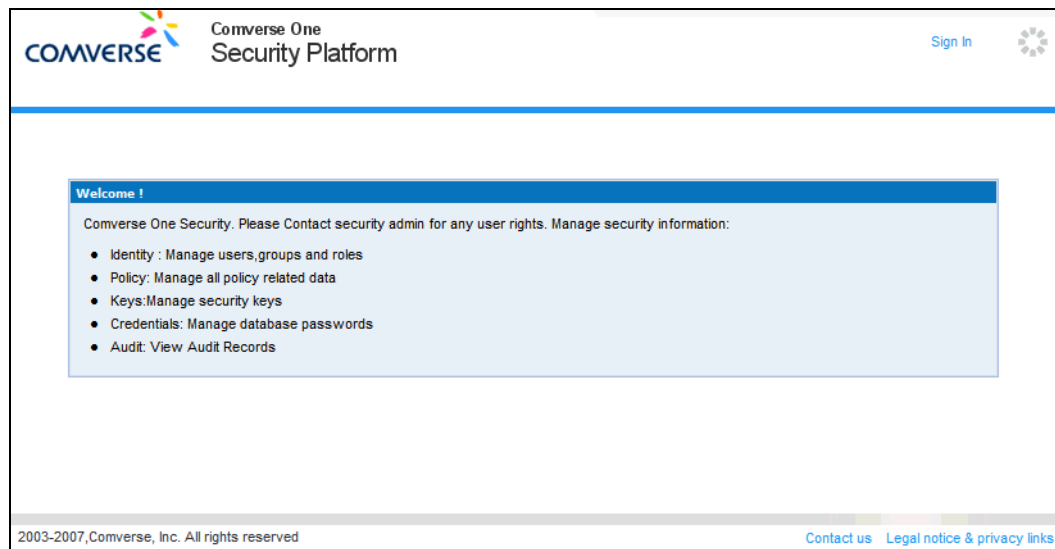
Logging In to the Security GUI

To log in to the Security GUI, do the following steps:

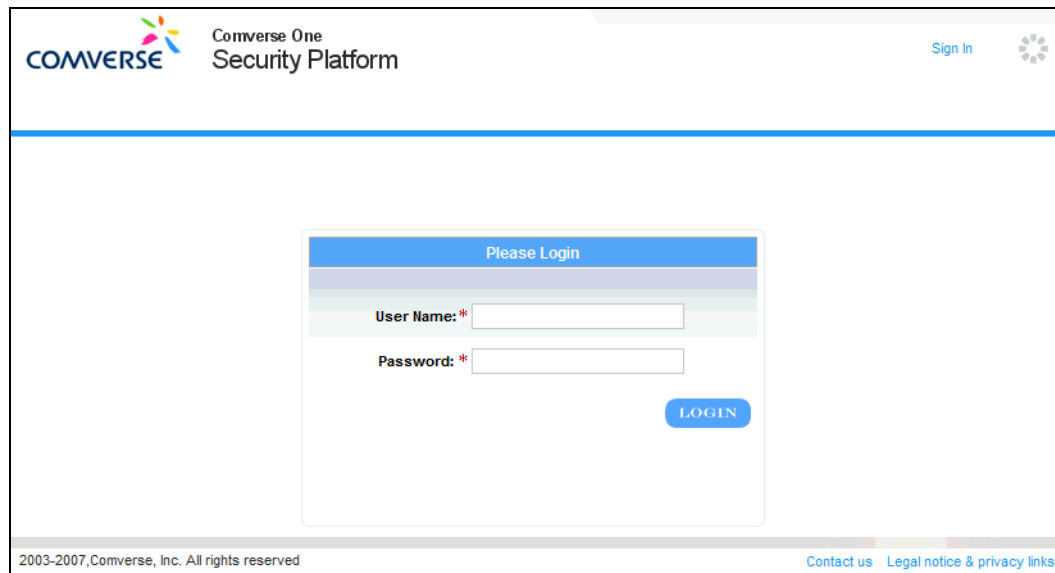
1. In your Web browser, go to the URL for the GUI.

An example of the URL is `http://<IP address>:8800/security/`, where *<IP address>* is the IP address for the Security Server. (This is the same as the IP address for the UPM if they are located on the same machine.) The introductory Welcome page appears, as shown in [Figure 43](#).

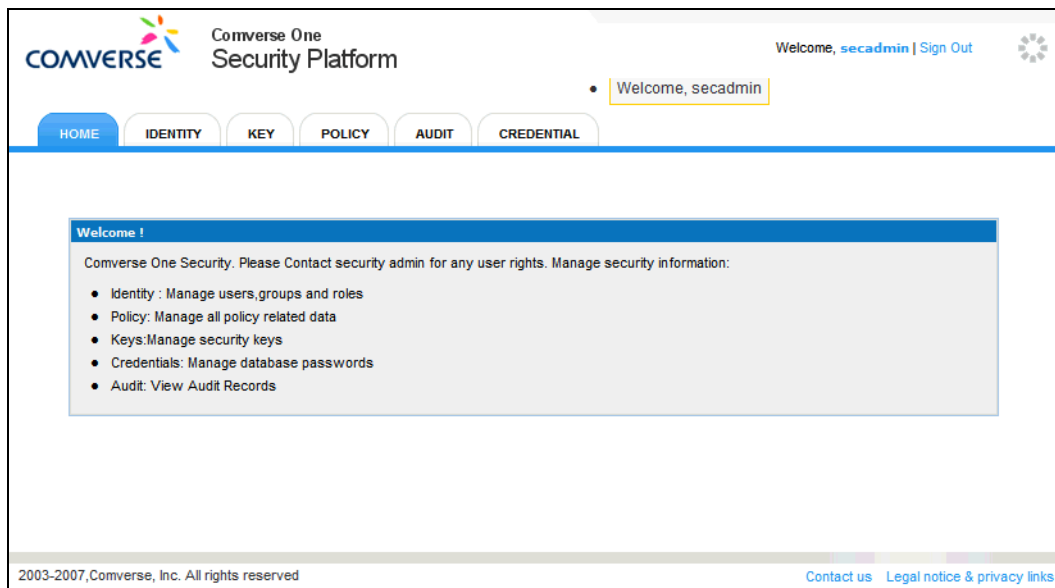
Figure 43 Security GUI — Introductory Welcome Page



2. At the top of the Welcome page, click the **Sign In** link.
The Login page appears, as shown in [Figure 44 on page 76](#).

Figure 44 Security GUI — Login Page

3. On the Login page, type your user name and password and click **Login**.
After your identity is authenticated, the Home page appears, as shown in [Figure 45](#).

Figure 45 Security GUI — Home Page

To navigate among the different functional areas, select one of the following top-level tabs that appear on all pages in the GUI:

- **Home:** Returns to the Home page.
- **Identity:** Provides access to identity management operations, including those for security realms, security realm groups, security roles, and users (defined per security realm). An operation to search for users across security realms is also available.
- **Key:** Provides access to key management operations, which deal with the centrally managed symmetric encryption keys.

- **Policy:** Provides access to policy management operations, including those for authorization policies and authorization rules.
- **Audit:** Provides read-only access to audit records.
- **Credential:** Provides access to credentials management operations. Currently, credentials include database passwords for business databases used by applications and SNMP community strings for network devices.

Logging Out of the Security GUI

To log out, click the **Sign Out** link that appears at the top of all pages in the Security GUI.



NOTE

Because of space considerations, figures in the remainder of this chapter focus only on areas of the Security GUI pages relevant to each discussion. For that reason, the **Sign Out** link that appears at the top of each GUI page is not included in later figures.

Identity Management

Identity management operations enable you to work with the following:

- **User Accounts:** User accounts are defined per security realm, and all operations on a user account occur within the context of the realm in which the user is provisioned.
- **Security Realm Groups:** For information on this concept, see [“What Are Security Realm Groups?” on page 17](#).
- **Security Roles:** For information on this concept, see [“What Are Security Roles?” on page 22](#).
- **Security Realms:** For information on this concept, see [“What Are Security Realms?” on page 17](#).

To go to the Identity Management area, select the top-level **Identity** tab on any page in the Security GUI. The Identity Management page appears. [Figure 46](#) shows an example of the page, with the UPSEC security realm selected. The Users tab lists all users defined for that realm.

Figure 46 Security GUI — Identity Management Page

The screenshot shows the Security GUI Identity Management page. The top navigation bar includes tabs for HOME, IDENTITY (selected), KEY, POLICY, AUDIT, and CREDENTIAL. Below this, there are sub-tabs for Users, Groups, Roles, and Realms. The Users tab is active, displaying a table of users for the selected UPSEC realm. On the left, a navigation tree lists various security realms, with UPSEC highlighted. At the bottom right of the table, there is an 'Add User' button.

User Name	First Name	Last Name	Realm	Email	Lock	Department	Phone	Force ChangePassword	LastUpdated	Reset Password	Operation
comms_owner	comms	owner	UPSEC		No	DWH	999-999-9999	No	04/02/2009 5:44:35 PM	Reset Password	✗
dwhuser	dwh_fn	dwh_ln	UPSEC	DWHUSER	No	DWH	999-999-9999	No	04/02/2009 5:44:35 PM	Reset Password	✗
secadmin	Admin_FN	Admin_LN	UPSEC	~	No	dept001	(856) 608-7407	No	04/14/2009 2:56:36 PM	Reset Password	✗
secuser	Sec_Admin_Fn	Sec_Admin_Ln	UPSEC		No	Engineering	999-999-9999	No	04/02/2009 5:44:36 PM	Reset Password	✗
upuser	UP_Admin_Fn	UP_Admin_Ln	UPSEC		No	Engineering	999-999-9999	No	04/02/2009 5:44:35 PM	Reset Password	✗
wktpuser	wktp_fn	wktp_ln	UPSEC		No	WKP	999-999-9999	No	04/02/2009 5:44:36 PM	Reset Password	✗

On the left side of the Identity Management page is a navigation tree of currently defined security realms. The Users, Groups, Roles, and Realms tabs on the page enable you to work with user accounts in a selected security realm, groups in a selected security realm, security roles (independent of security realms), and security realms themselves.

Select a security realm in the tree before selecting a tab for the following:

- Users
- Groups

Working with User Accounts

The following sections provide information on working with user accounts. User accounts are defined per security realm.

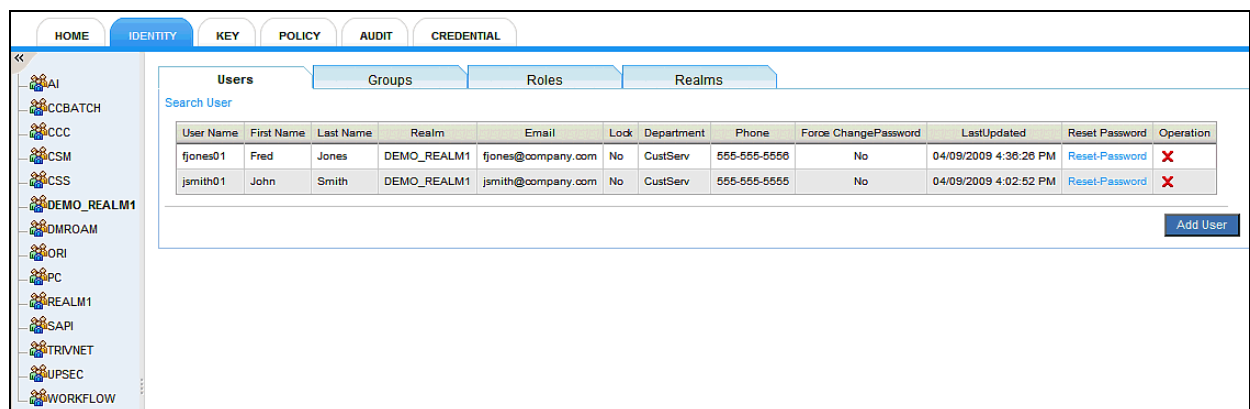
Viewing User Accounts in a Security Realm

To view all user accounts defined for a security realm, do the following steps:

1. Select the top-level **Identity** tab on any page in the Security GUI.
2. On the Identity Management page that appears, select the appropriate security realm from the realm navigation tree.

The Users tab on the page shows all users defined for the selected realm. [Figure 47](#) shows an example.

Figure 47 Security GUI — Users Tab Showing Users for a Realm



The screenshot shows the Security GUI interface. The top navigation bar includes tabs: HOME, IDENTITY (selected), KEY, POLICY, AUDIT, and CREDENTIAL. On the left is a navigation tree with various security realms, including DEMO_REALM1 which is selected. The main content area is titled 'Users' and contains a 'Search User' input field. Below this is a table with the following columns: User Name, First Name, Last Name, Realm, Email, Lock, Department, Phone, Force Change Password, Last Updated, Reset Password, and Operation. Two users are listed in the table. At the bottom right of the table area is an 'Add User' button.

User Name	First Name	Last Name	Realm	Email	Lock	Department	Phone	Force Change Password	Last Updated	Reset Password	Operation
fjones01	Fred	Jones	DEMO_REALM1	fjones@company.com	No	CustServ	555-555-5555	No	04/09/2009 4:38:28 PM	Reset Password	
jsmith01	John	Smith	DEMO_REALM1	jsmith@company.com	No	CustServ	555-555-5555	No	04/09/2009 4:02:52 PM	Reset Password	

From the Users tab, you can do the following operations on user accounts:

- **Reset a User Password:** Click the **Reset Password** link in the row for the user. A message appears on the screen indicating that the operation was successful and displaying the new password. The user will be required to change this password at the next login.
- **View/Modify All Data for a User Account:** To view/modify data for the user account, click the row for the user. (For details on modifications, see [“Modifying User Accounts.”](#))
- **Delete a User Account from the Security Realm:** Click the Delete (red X) button in the Operation column for the user you want to delete. For more information, see [“Deleting User Accounts” on page 80.](#)
- **Add a New User Account to the Security Realm:** Click **Add User**. For more information, see [“Creating User Accounts” on page 80.](#)
- **Search for Users across Realms:** To do this, click the **Search User** link. For more information, see [“Searching for Users across Security Realms” on page 83.](#)

Modifying User Accounts

To modify a user account, do the following steps:

1. View user accounts for the appropriate security realm. To do this, follow instructions in [“Viewing User Accounts in a Security Realm.”](#)

2. On the Users tab that lists all users for the realm, click the row for the user.

A new page appears, with the Edit User Details tab selected by default. [Figure 48](#) shows an example.

Figure 48 Security GUI — Edit User Details Tab (Modifying a User)

3. Make any needed modifications on the Edit User Details tab.

For descriptions of fields on the tab, see [Table 5, “Add User Details Tab — Field Descriptions,” on page 82](#). Although that table is for adding user details, it contains field descriptions relevant to editing user details.

4. After finishing modifications on the Edit User Details tab, click **Update**.

A message appears on the screen indicating that the operation was successful.

5. To modify custom attributes defined for the user, select the User Attributes tab to display the currently defined attributes. [Figure 49](#) shows an example.

Figure 49 Security GUI — User Attributes Tab, Current Attributes (Modifying a User)

Name	Value	Remove
nine	9	<input type="checkbox"/>
ten	10	<input type="checkbox"/>

6. To delete any current custom attributes for the user, do the following:

- a. On the User Attributes tab, select the checkbox for the attribute in the Remove column.
- b. Click **Delete Attribute**.

A message appears on the screen indicating that the operation was successful. The page is refreshed to show the current attributes.

**NOTE**

No incomplete rows should be visible in the Add New User Attributes area when you click **Delete Attribute**. Otherwise, a validation error will be displayed.

7. To add new custom attributes for the user, do the following:
 - a. On the User Attributes tab, click the + (plus) icon beside the Add New User Attributes label. Doing this adds a row of attribute definition fields each time you click the icon. [Figure 50](#) shows an example. Click the - (minus) icon to remove a row.

Figure 50 Security GUI — User Attributes Tab, Add Attributes (Modifying a User)

- b. Type entries in the Attribute Name and Attribute Value fields.
- c. Click **Save Attributes**.
A message appears on the screen indicating that the operation was successful. The page is refreshed to show the current attributes.

Deleting User Accounts

Two methods are available to delete a user account from a security realm.

For the first method, do the following steps:

1. View user accounts for the appropriate security realm. To do this, follow instructions in [“Viewing User Accounts in a Security Realm” on page 78](#).
2. On the Users tab that lists the users defined for the security realm, click the Delete (red X) button in the Operation column for the user whose account you want to delete.
A dialog box appears, asking you to confirm the deletion.
3. Click **OK** in the dialog box.
A message appears on the screen indicating that the operation was successful.

For the second method, do the following steps:

1. View user accounts for the appropriate security realm. To do this, follow instructions in [“Viewing User Accounts in a Security Realm” on page 78](#).
2. On the Users tab that lists the users defined for the security realm, click the row for the user.
3. On the Edit User Details tab that shows details for the selected user, click **Delete**.

Creating User Accounts

User accounts are defined per security realm.

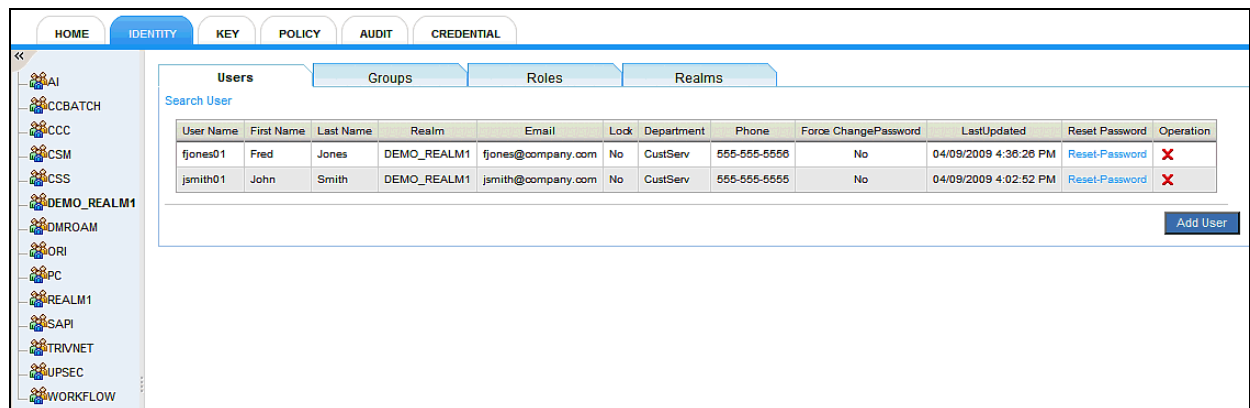
To create user accounts, do the following steps:

1. Select the top-level **Identity** tab on any page in the Security GUI.
2. On the Identity Management page that appears, select the appropriate security realm from the realm navigation tree.

Users currently defined for the selected security realm are displayed on the Users tab.

[Figure 51](#) shows an example.

Figure 51 Security GUI — Users Tab (Creating a User)



3. To create a user account in the selected security realm, do one of the following: (1) click **Add User** on the Users tab or (2) right-click the realm name in the realm navigation tree and select **Add User** from the popup menu.

A tabbed page appears where you can define all needed information for the user account. The Add User Details tab is selected by default. [Figure 52](#) shows an example.

Figure 52 Security GUI — Add User Details Tab (Creating a User)

The screenshot shows the 'Add User Details' tab in the Security GUI. It contains various input fields for user information, with red asterisks indicating required fields.

User Attributes

Fields include:

- User Name: *
- First Name: *
- Phone:
- Email: *
- Groups: (List box with DEMO_GROUP2, DEFAULT_GROUP_DEMO_REALM1, DEMO_GROUP1)
- Password: *
- Middle Name:
- Extension:
- Lock: (Dropdown menu with -No- selected)
- Re-Password: *
- Last Name: *
- Department:
- Force Change Password: (Dropdown menu with -No- selected)
- Priority Group: (Dropdown menu)

Buttons for 'Save' and 'Cancel' are at the bottom right.

4. Fill in data for the fields on the Add User Details tab (red asterisks indicate required fields). [Table 5, "Add User Details Tab — Field Descriptions," on page 82](#) describes the fields on the tab.

Table 5 Add User Details Tab — Field Descriptions

Field	Description
User Name	User ID used to log in to the system.
Password	User's password, which must comply with the security realm's password policy.
Re-Password	Re-entry of the user's password.
First Name	User's first name.
Middle Name	User's middle name.
Last Name	User's last name.
Phone	User's phone number.
Extension	User's phone extension, if any.
Department	User's department.
Email	User's email address.
Lock	Indicates whether the user account is locked. A locked account prevents logins until it is unlocked.
Force Change Password	Indicates whether the user will be forced to change his/her password at the next login.
Groups	The list on the left shows the groups currently defined for the security realm. The list on the right shows the groups that the user belongs to. Click the >> and << icons to move the selected group(s) from one list to the other.
Priority Group	User's priority group, which is used to (1) resolve attribute conflicts and (2) establish soft timeout/hard timeout values for user sessions.

5. After filling in data on the Add User Details tab, click **Save**.

A message appears on the screen indicating that the operation was successful. The tab changes from Add User Details to Edit User Details to enable modifications.

**NOTE**

You must click **Save** on the Add User Details tab to create the basic user account. Then you can add data on the User Attributes tab to update the account.

6. To define custom attributes for the user, if needed, do the following:
- Select the User Attributes tab.
 - On the User Attributes tab, click the + (plus) icon beside the Add New User Attributes label on the User Attributes tab. Doing this adds a row of attribute definition fields each time you click the icon. [Figure 53 on page 83](#) shows an example. Click the - (minus) icon to remove a row.

Figure 53 Security GUI — User Attributes Tab, New Attributes (Creating a User)

The screenshot shows the 'User Attributes' tab in the Security GUI. Under the 'Add New User Attributes' section, there are two rows of input fields. Each row has an 'Attribute Name' field, an 'Attribute Value' field, and a 'Save' button. At the bottom of the section are 'Save Attributes' and 'Clear' buttons.

- c. In the Add New User Attributes area, type entries in the Attribute Name and Attribute Value fields.
- d. Click **Save Attributes**.
A message appears on the screen indicating that the operation was successful. The page is refreshed to show the current attributes. [Figure 54](#) shows an example.

Figure 54 Security GUI — User Attributes Tab, Current Attributes (Creating a User)

The screenshot shows the 'Current User Attributes' section. It contains a table with the following data:

Name	Value	Remove
five	5	<input type="checkbox"/>
four	4	<input type="checkbox"/>

Below the table are 'Delete Attribute' and 'Cancel' buttons.

7. To delete a currently defined user attribute, do the following:
 - a. On the User Attributes tab, select the checkbox in the Remove column.
 - b. Click **Delete Attribute**.
A message appears on the screen indicating that the operation was successful. The page is refreshed to reflect the current attributes.

**NOTE**

No incomplete rows should be visible in the Add New User Attributes area when you click **Delete Attribute**. Otherwise, a validation error will be displayed.

Searching for Users across Security Realms

Although users are defined per security realm, you can search across realms to find one or more users.

To search for users across security realms, do the following steps:

1. Select the top-level **Identity** tab on any page in the Security GUI.
2. On the Identity Management page that appears, on the Users tab, click the **Search User** link.

The area that normally displays the realm navigation tree changes to display search fields. [Figure 55 on page 84](#) shows the fields.

Figure 55 Security GUI — Fields to Search for Users across Realms

The screenshot shows a search form with the following fields and controls:

- User Name:** Text input field
- First Name:** Text input field
- Middle Name:** Text input field
- Last Name:** Text input field
- Email:** Text input field
- Lock Status:** Dropdown menu with "-Select-" selected
- Department:** Text input field
- Phone No:** Text input field
- Force Change Password:** Dropdown menu with "-Select-" selected
- Search User:** Blue button
- Return Realm View:** Blue link at the bottom left

3. Enter data in at least one field. It can be either complete data (for example, the complete last name) or the first few characters of the data. The search is case-insensitive.
4. Click **Search User**.
Users found by the search are listed on the Users tab. (A message appears on the screen if the search produced no results.) The display of users is similar to the display for users in a realm, as shown in [Figure 47 on page 78](#). An additional Realm column identifies the security realm for each user. You can do the same operations on the list of users as you can for a typical list of users displayed on the Users tab.
5. To remove the search fields and search results, click the **Return Realm View** link below the search fields, shown in [Figure 55](#).

The realm navigation tree appears in place of the search fields. The list of users on the Users tab changes to show users in the realm currently selected in the realm tree.

Working with Security Realm Groups

The following sections provide information on working with groups in a security realm. For background information on groups, see [“What Are Security Realm Groups?” on page 17](#).

Viewing Groups

To view groups defined for a security realm, do the following steps:

1. Select the top-level **Identity** tab on any page in the Security GUI.
2. On the Identity Management page that appears, (1) select the appropriate security realm from the navigation tree and (2) select the Groups tab.

The Groups tab lists all groups defined for the selected security realm. [Figure 56 on page 85](#) shows an example.

Figure 56 Security GUI — Groups Tab Showing Groups for a Realm

Group Name	Short Description	Description	Default	LangCode	Roles	LastUpdated	Operation
CSM_ADMIN	Admin Group	Admin Group	No		ADMIN	08/05/2010 5:45:50 PM	X
CSM_SUPER	Senior Group	Senior Group	No		CSMSUPERUSER	08/05/2010 5:45:50 PM	X
CSM_USER	Junior Group	Junior Group	No		CSMUSER	08/05/2010 5:45:50 PM	X
DEFAULT_GROUP_CSM		default group for DEFAULT_GROUP_CSM	Yes		GUEST,ADMIN	08/05/2010 5:45:50 PM	X

From the Groups tab, you can do the following operations on groups:

- **View/Modify All Data for a Group:** To view / modify data for a group, click the row for the group. For details on modifications, see [“Modifying Groups.”](#)
- **Delete a Group:** Click the Delete (red X) button in the Operation column for the group you want to delete. For information on the effect that deleting a group has on users who belong to the group, see [“Deleting Groups” on page 87.](#)
- **Add a New Group:** Click **Add Group**. For information, see [“Creating Groups” on page 87.](#)

Modifying Groups

To modify a group, do the following steps:

1. View groups for the appropriate security realm. To do this, follow instructions in [“Viewing Groups” on page 84.](#)
2. Click the row for the group you want to modify.

A new tabbed page appears, with the Edit Group Details tab selected by default. [Figure 57](#) shows an example.

Figure 57 Security GUI — Edit Group Details Tab (Modifying a Group)

Edit Group Details **Group Attributes**

Group Name: * Short Description:

Description: Realm Name:

Is Default Group: ☒ Yes

Roles:

ACC_ADD_SUBSCRIBER
 ACC_CHANGE_BALANCE
 ACC_CHANGE_LR
 ACC_CHANGE_OFFER
 ACC_CHANGE_PLANS
 ACC_CHANGE_TERMS
 ACC_DEACTIVATE
 ACC_MOVE_SUBSCRIBER
 ACC_REACTIVATE
 ACC_SUBSCRIBE_OFFER
 ACC_UNSUBSCRIBE_OFFER
 ACC_VIEW_BALANCE

ADMIN
 GUEST

Soft Timer: * Hard Timer: * Is Default Policy: ☒ Yes

3. Make any needed modifications on the Edit Group Details tab.
For descriptions of fields on the Edit Group Details tab, see [Table 6, “Add Group Details Tab — Field Descriptions,” on page 88](#). Although that table is for adding group details, it contains field descriptions relevant to editing group details.
4. After finishing modifications on the Edit Group Details tab, click **Update**.
A message appears on the screen indicating that the operation was successful.
5. To modify custom attributes defined for the group, select the Group Attributes tab to display the currently defined attributes. [Figure 58](#) shows an example.

Figure 58 Security GUI — Group Attributes Tab, Current Attributes (Modifying a Group)

The screenshot shows the Security GUI with the 'IDENTITY' tab selected. Under the 'IDENTITY' tab, the 'Group Attributes' sub-tab is active. At the top, there are tabs for 'HOME', 'IDENTITY', 'KEY', 'POLICY', 'AUDIT', and 'CREDENTIAL'. Below these, there are two sub-tabs: 'Edit Group Details' and 'Group Attributes'. The 'Group Attributes' sub-tab is selected. Below the sub-tabs, there is a section titled 'Add New Group Attributes' with a plus icon. Below that, there is a section titled 'Current Group Attributes' with a table. The table has three columns: 'Name', 'Value', and 'Remove'. There are two rows of attributes: 'seven' with value '7' and 'six' with value '6'. Each row has a checkbox in the 'Remove' column. At the bottom of the table, there are two buttons: 'Delete Attribute' and 'Cancel'.

Name	Value	Remove
seven	7	<input type="checkbox"/>
six	6	<input type="checkbox"/>

6. To delete any current custom attributes for the group, do the following:
 - a. On the Group attributes tab, select the checkbox for the attribute in the Remove column.
 - b. Click **Delete Attribute**.
A message appears on the screen indicating that the operation was successful. The page is refreshed to show the current attributes.



NOTE No incomplete rows should be visible in the Add New Group Attributes area when you click **Delete Attribute**. Otherwise, a validation error will be displayed.

7. To add new custom attributes for the group, do the following:
 - a. On the Group Attribute tab, click the + (plus) icon beside the Add New Group Attributes label. Doing this adds a row of attribute definition fields each time you click the icon. [Figure 59 on page 87](#) shows an example. Click the - (minus) icon to remove a row.

Figure 59 Security GUI — Group Attributes Tab, Add Attributes (Modifying a Group)

The screenshot shows the Security GUI with tabs for HOME, IDENTITY, KEY, POLICY, AUDIT, and CREDENTIAL. The IDENTITY tab is selected, and the 'Group Attributes' sub-tab is active. Below the sub-tabs, there is a section titled 'Add New Group Attributes' with a plus icon. This section contains two rows of input fields. Each row has an 'Attribute Name:' field followed by an 'Attribute Value:' field, with a small blue minus icon to the right of each value field. Below these fields are two buttons: 'Save Attributes' and 'Cancel'. At the bottom of the form, there is a section titled 'Current Group Attributes' with a 'Cancel' button.

- b. To add attributes, type entries in the Attribute Name and Attribute Value fields.
- c. Click **Save Attributes**.

A message appears on the screen indicating that the operation was successful. The page is refreshed to show the current attributes.

Deleting Groups



NOTE

Before deleting a group that is the priority group for any users, it is recommended that you reassign those users to a different priority group.

Deleting a group means that any group attributes and roles, which were previously inherited by users belonging to the group, will no longer be inherited by those users. If the deleted group was the priority group for any users, those users will be reassigned to the realm's DEFAULT group as their priority group. (The users will then inherit the DEFAULT group's roles, which are ADMIN and GUEST, and the soft timeout/hard timeout values of the DEFAULT group.)

To delete a group from a security realm, do the following steps:

1. View groups for the appropriate security realm. To do this, follow instructions in [“Viewing Groups” on page 84](#).
2. On the Groups tab that lists the groups defined for the security realm, click the Delete (red X) button in the Operation column for the group that you want to delete.
A dialog box appears, asking you to confirm the deletion.
3. Click **OK** in the dialog box.
A message appears on the screen indicating that the operation was successful.

Creating Groups

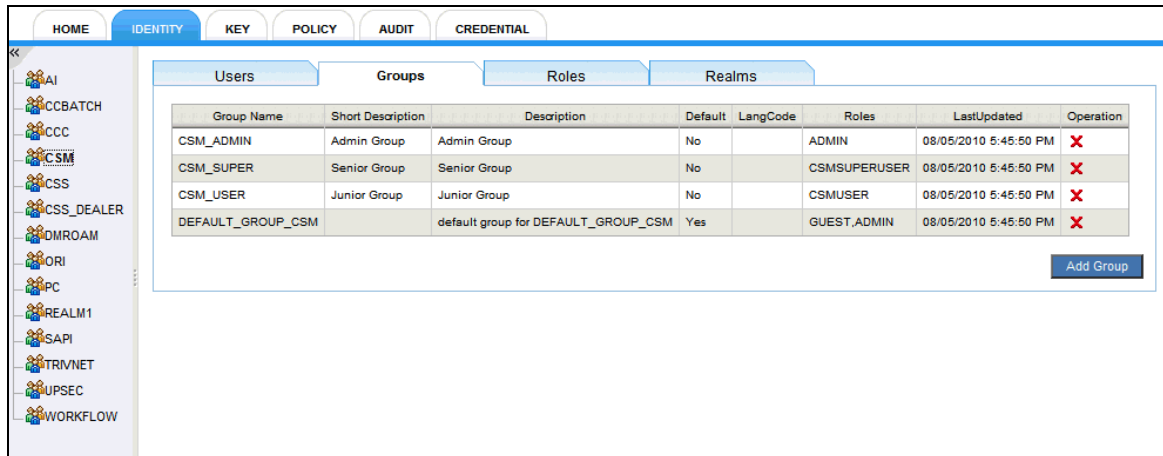
Groups are defined per security realm. For further details about groups, see [“What Are Security Realm Groups?” on page 17](#).

To create groups, do the following steps:

1. Select the top-level **Identity** tab on any page in the Security GUI.
2. On the Identity Management page that appears, select the appropriate security realm from the realm navigation tree.

3. Do one of the following:
 - a. On the Identity Management page, select the Groups tab. This tab lists all groups currently defined for the selected security realm, as shown in [Figure 60](#). Click **Add Group** on the Groups tab.

Figure 60 Security GUI — Groups Tab (Creating a Group)



- b. As an alternative to step 3a, right-click the realm name in the realm navigation tree and select **Add Group** from the popup menu.

After you do step 3a or 3b, a tabbed page appears where you can define all needed information for the group. The Add Group Details tab is selected by default. [Figure 61](#) shows an example.

Figure 61 Security GUI — Add Group Details Tab (Creating a Group)

4. Fill in data for the fields on the Add Group Details tab (red asterisks indicate required fields). [Table 6](#) describes the fields.

Table 6 Add Group Details Tab — Field Descriptions

Field	Description
Group Name	Unique group name within the security realm.
Short Description	Short description of the group.
Description	Description of the group.

Table 6 Add Group Details Tab — Field Descriptions (Continued)

Field	Description
Realm Name	Name of the security realm for which the group is being created. (The field is auto-filled by the selection from the security realm navigation tree.)
Roles	The list on the left shows all the currently defined security roles. The list on the right shows the roles currently associated with the group. Click the >> and << icons to move the selected role(s) from one list to the other. If you have not yet defined the roles that you want to associate with the group, you can modify the group later to add the roles.
Soft Timer	Soft timeout, which is the maximum period of session inactivity (in milliseconds) for users in the group, after which a user is logged out and must log in again. The value for soft timeout must be less than the value for hard timeout.
Hard Timer	Hard timeout, which is the maximum session duration (in milliseconds) for users in the group, after which a user is logged out and must log in again. The value for hard timeout must be greater than the value for soft timeout.

5. After filling in data on the Add Group Details tab, click **Save**.

A message appears on the screen indicating that the operation was successful. The tab changes from Add Group Details to Edit Group Details to enable modifications.

**NOTE**

You must click **Save** on the Add Group Details tab to create the group. Then you can add data on the Group Attributes tab to update the group.

6. To define custom attributes for the group, if needed, do the following:
- Select the Group Attributes tab.
 - On the Group Attributes tab, click the + (plus) icon beside the Add New Group Attributes label. Doing this adds a row of attribute definition fields each time you click the icon. [Figure 62](#) shows an example. Click the - (minus) icon to remove a row.

Figure 62 Security GUI — Group Attributes Tab, Add Attributes (Creating a Group)

- Type entries in the Attribute Name and Attribute Value fields.

- d. Click **Save Attributes**.

A message appears on the screen indicating that the operation was successful. The page is refreshed to reflect the current attributes. [Figure 63](#) shows an example.

Figure 63 Security GUI — Group Attributes Tab, Current Attributes (Creating a Group)

The screenshot shows the Security GUI with the 'IDENTITY' tab selected. Under 'Group Attributes', there is a section for 'Current Group Attributes'. It contains a table with two rows: 'seven' with value '7' and 'six' with value '6'. Each row has a 'Remove' checkbox. Below the table are 'Delete Attribute' and 'Cancel' buttons.

Name	Value	Remove
seven	7	<input type="checkbox"/>
six	6	<input type="checkbox"/>

7. To delete a currently defined group attribute, do the following:

- On the Group Attributes tab, select the checkbox for the attribute in the Remove column.
- Click **Delete Attribute**.

A message appears on the screen indicating that the operation was successful. The page is refreshed to show the current attributes.



NOTE

No incomplete rows should be visible in the Add New Group Attributes area when you click **Delete Attribute**. Otherwise, a validation error will be displayed.

Working with Security Roles

The following sections provide information on security roles. Roles are independent of security realms. For background information on roles, see [“What Are Security Roles?” on page 22](#).

Viewing Roles

To view currently defined roles, do the following steps:

- Select the top-level **Identity** tab on any page in the Security GUI.
- On the Identity Management page that appears, select the Roles tab.

The Roles tab shows a table of the currently defined roles. [Figure 64 on page 91](#) shows an example of the tab.

Figure 64 Security GUI — Roles Tab

Role Name	Short Description	Description	LastUpdated	Operation
ACC_ADD_SUBSCRIBER			04/02/2009 5:44:35 PM	X
ACC_CHANGE_BALANCE			04/02/2009 5:44:35 PM	X
ACC_CHANGE_LR			04/02/2009 5:44:34 PM	X
ACC_CHANGE_OFFER			04/02/2009 5:44:35 PM	X
ACC_CHANGE_PLANS			04/02/2009 5:44:35 PM	X
ACC_CHANGE_TERMS			04/02/2009 5:44:35 PM	X
ACC_DEACTIVATE			04/02/2009 5:44:35 PM	X
ACC_MOVE_SUBSCRIBER			04/02/2009 5:44:35 PM	X
ACC_REACTIVATE			04/02/2009 5:44:35 PM	X

Adding Roles

To add security roles, do the following steps:

1. View roles as described in [“Viewing Roles” on page 90](#).
2. On the Roles tab (shown in [Figure 64](#)), type data in fields for the Role Name and, optionally, Short Description and Description.
3. Click **Add Role**.

A message appears on the screen indicating that the operation was successful.

Deleting Roles

To delete a role, do the following steps:

1. View roles as described in [“Viewing Roles” on page 90](#).
2. On the Roles tab (shown in [Figure 64](#)), click the Delete (red X) button in the Operation column for the role you want to delete.
A dialog box appears, asking you to confirm the deletion.
3. Click **OK** in the dialog box.

A message appears on the screen indicating that the operation was successful.

Working with Security Realms

The following sections provide information on security realms. For background information on realms, see [“What Are Security Realms?” on page 17](#).

Viewing Realms

To view currently defined security realms, do the following steps:

1. Select the top-level **Identity** tab on any page in the Security GUI.
2. On the Identity Management page that appears, select the Realms tab.
The Realms tab shows all the currently defined security realms. [Figure 65 on page 92](#) shows an example of the tab.

Figure 65 Security GUI — Realms Tab

Realm Name	Short Description	Description	Operation
AI	AppIntegrator	AppIntegrator	X
CCBATCH	--	--	X
CCC	--	--	X
CSM	CSM	CSM	X
CSS	SelfCare	SelfCare	X
DEMO_REALM1	DemoRealm1	Demo Realm 1	X
DMROAM	DMROAM	DMROAM	X
ORI	ORI	ORI	X
PC	ProductCatalog	ProductCatalog	X
REALM1	Realm1	Realm1	X
SAPI	SAPI	SAPI	X
TRIVNET	Trivnet	Trivnet	X
UPSEC	SRI	UPSEC	X
WORKFLOW	WORKFLOW	WORKFLOW	X

From the Realms tab, you can do the following operations on security realms:

- **View/Modify All Data for a Realm:** To view /modify data for a realm, click the row for the realm. (For details on modifications, see [“Modifying Realms.”](#))
- **Delete a Realm:** To delete a security realm, click the Delete (red X) button in the Operation column for the realm you want to delete.



Deleting a security realm deletes all user accounts and realm groups defined for the realm, along with the realm itself. Carefully consider whether you want to delete a security realm.

- **Add a New Realm:** For information, see [“Creating Realms” on page 95.](#)

Modifying Realms

To modify a realm, do the following steps:

1. View the currently defined security realms, as described in [“Viewing Realms” on page 91.](#)
2. Click the row for the realm you want to modify.

A new tabbed page appears, with the Edit Realm tab selected by default. [Figure 66 on page 93](#) shows an example.

Figure 66 Security GUI — Edit Realm Tab (Modifying a Realm)

3. On the Edit Realm tab, make any needed modifications. (Modifications are permitted only to the Short Description and Description fields.)
4. After finishing the modifications, click **Update**.
A message appears on the screen indicating that the operation was successful.
5. To modify custom attributes defined for the realm, select the Realm Attributes tab to display the currently defined attributes. [Figure 67](#) shows an example.

Figure 67 Security GUI — Realm Attributes Tab, Current Attributes (Modifying a Realm)

Name	Value	Remove
one	1	<input type="checkbox"/>
two	2	<input type="checkbox"/>

6. To delete any current custom attributes for the realm, do the following:
 - a. On the Realm Attributes tab, select the checkbox for the attribute in the Remove column.
 - b. Click **Delete Attribute**.
A message appears on the screen indicating that the operation was successful. The page is refreshed to show the current attributes.

**NOTE**

No incomplete rows should be visible in the Add New Realm Attributes area when you click **Delete Attribute**. Otherwise, a validation error will be displayed.

7. To add new custom attributes for the realm, do the following:
 - a. On the Realm Attributes tab, click the + (plus) icon beside the Add New Realm Attributes label. Doing this adds a row of attribute definition fields each time you click the icon. [Figure 68 on page 94](#) shows an example. Click the - (minus) icon to remove a row.

Figure 68 Security GUI — Realm Attributes Tab, Add Attributes (Modifying a Realm)

The screenshot shows the 'Add New Realm Attributes' section of the Security GUI. It features two rows of input fields for 'Attribute Name' and 'Attribute Value', each with a red asterisk indicating a required field. Below these fields are 'Save Attributes' and 'Cancel' buttons. The 'Current Realm Attributes' section below shows a table with two existing attributes: 'one' with value '1' and 'two' with value '2'. Each row has a 'Remove' checkbox. At the bottom are 'Delete Attribute' and 'Cancel' buttons.

Name	Value	Remove
one	1	<input type="checkbox"/>
two	2	<input type="checkbox"/>

- b. Type entries in the Attribute Name and Attribute Value fields.
 - c. Click **Save Attributes**.

A message appears on the screen indicating that the operation was successful. The page is refreshed to show the current attributes.
8. To modify the current password policy for the security realm, do the following:
 - a. Select the Password Policy tab.

The Password Policy tab shows all parts of the currently defined password policy. [Figure 69](#) shows an example.

Figure 69 Security GUI — Password Policy Tab (Modifying a Realm)

The screenshot shows the 'Password Policy' tab in the Security GUI. It contains various input fields for password requirements: Minimum Alpha (1), Minimum Other (1), Minimum Length (4), Minimum Difference (4), Minimum Age (0), Maximum Age (12), Maximum Expired (2), History Expire (3), History Size (5), Maximum Length (20), Maximum Retries (3), Lock Interval (30), Dictionary List (empty), and IsDefault Policy (No). At the bottom right are 'Update' and 'Cancel' buttons.

- b. Make any needed modifications on the Password Policy tab. For descriptions of fields on the tab, see [Table 7, “Password Policy Tab — Field Descriptions,” on page 98](#).



If user accounts have been created for the realm, be cautious in changing certain portions of the policy, such as minimum password length, minimum number of alphabetic characters, or minimum number of other characters. Otherwise, passwords for user accounts defined for the realm might no longer be valid.

- c. After finishing your modifications, click **Update**.
A message appears on the screen indicating that the operation was successful.

Deleting Realms



Deleting a security realm deletes all user accounts and realm groups defined for the realm, along with the realm itself. Carefully consider whether you want to delete a security realm.

Two methods are available to delete security realms.

For the first method, do the following steps:

1. View security realms. To do this, follow instructions in [“Viewing Realms” on page 91](#).
2. On the Realms tab that lists the currently defined security realms, click the Delete (red X) button in the Operation column for the realm you want to delete.
A dialog box appears, asking you to confirm the deletion.
3. Click **OK** in the dialog box.
A message appears on the screen indicating that the operation was successful.

For the second method, do the following steps:

1. View security realms. To do this, follow instructions in [“Viewing Realms” on page 91](#).
2. On the Realms tab that lists the currently defined security realms, click the row for the realm.
3. On the Edit Realm tab that displays details for the selected realm, click **Delete**.

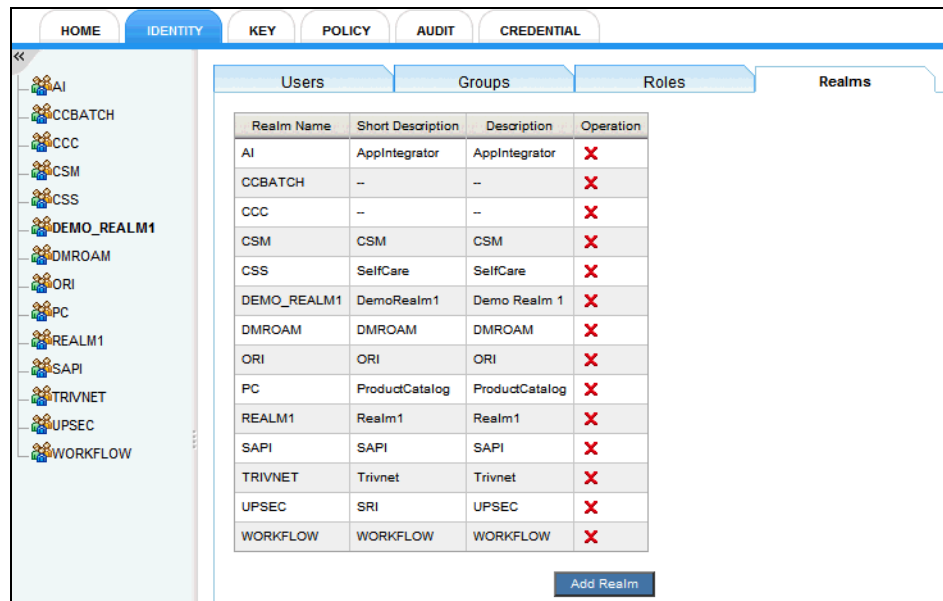


The UPSEC security realm cannot be deleted. This realm contains the administrator user and users who can perform actions on the Security Server and Unified Platform. Deletion of this realm is not allowed because it would destroy the administrator account.

Creating Realms

To create security realms, do the following steps:

1. Select the top-level **Identity** tab on any page in the Security GUI.
2. On the Identity Management page that appears, select the Realms tab.
All currently defined security realms are displayed on the tab. [Figure 70 on page 96](#) shows an example.

Figure 70 Security GUI — Realms Tab (Creating a Realm)

3. On the Realms tab, click **Add Realm**.

A tabbed page appears where you can define all needed information for the realm. The Add Realm tab is selected by default. [Figure 71](#) shows an example.

Figure 71 Security GUI — Add Realm Tab (Creating a Realm)

The screenshot shows the 'Add Realm' tab in the Security GUI. It contains three input fields: 'Realm Name' (marked with a red asterisk), 'Short Description', and 'Description'. The 'Description' field has a small icon to its right. At the bottom right, there are 'Save' and 'Cancel' buttons.

4. Fill in data for the fields on the Add Realm tab (red asterisks indicate required fields).
5. After filling in data, click **Save**.

A message appears on the screen indicating that the operation was successful. The tab changes from Add Realm to Edit Realm to enable modifications.

**NOTE**

You must click **Save** on the Add Realm tab to create the basic realm. Then you can add data on the Realm Attributes and Password Policy tabs to update the realm.

6. To define custom attributes for the realm, if needed, do the following:
 - a. Select the Realm Attributes tab.
 - b. On the Realm Attributes tab, click the + (plus) icon beside the Add New Realm Attributes label. Doing this adds a row of attribute definition fields each time you click the icon. [Figure 72 on page 97](#) shows an example. Click the - (minus) icon to remove a row.

Figure 72 Security GUI — Realm Attributes Tab, New Attributes (Creating a Realm)

- c. Type entries in the Attribute Name and Attribute Value fields
- d. Click **Save Attributes**.

A message appears on the screen indicating that the operation was successful. The page is refreshed to show the current attributes. [Figure 73](#) shows an example.

Figure 73 Security GUI — Realm Attributes Tab, Current Attributes (Creating a Realm)

Name	Value	Remove
one	1	<input type="checkbox"/>
two	2	<input type="checkbox"/>

7. To delete a currently defined realm attribute, do the following:
 - a. On the Realm Attributes tab, select the checkbox for the attribute in the Remove column.
 - b. Click **Delete Attribute**.

A message appears on the screen indicating that the operation was successful. The page is refreshed to show the current attributes.

**NOTE**

No incomplete rows should be visible in the Add New Realm Attributes area when you click **Delete Attribute**. Otherwise, a validation error will be displayed.

8. To define a custom password policy for the security realm, do the following:
 - a. Select the Password Policy tab.

For a newly created realm, fields on the Password Policy tab are automatically filled with values taken from the default password policy. [Figure 74 on page 98](#) shows an example of the tab, with default values.

Figure 74 Security GUI — Password Policy Tab (Creating a Realm)

- b. Make any needed modifications on the Password Policy tab. [Table 7](#) provides field descriptions.
- c. After finishing the modifications, click **Update**
A message appears on the screen indicating that the operation was successful.

**NOTE**

The passwords for all user accounts later defined for the security realm must comply with the realm's password policy.

Table 7 Password Policy Tab — Field Descriptions

Field	Description
Minimum Alpha	Minimum number of alphabetic characters.
Minimum Other	Minimum number of other characters, which can be numbers or special characters such as #, &, %, and so on.
Minimum Length	Minimum password length.
Minimum Difference	Minimum number of characters in a new password that must be different from characters in the old password.
Minimum Age	Minimum password age, in weeks, before a password can be changed.
Maximum Age	Maximum password age, in weeks, at which time the user is notified to change the password.
Maximum Expired	Maximum expiration threshold, which is the maximum number of weeks beyond the maximum password age that a password can be changed by the user.
History Expire	Number of weeks in which a user cannot reuse a previous password.
History Size	Number of previous passwords that a user cannot reuse.
Maximum Length	Maximum password length.
Maximum Retries	Maximum number of consecutive failed login attempts (maximum retries) allowed before the user account is locked, disabling logins.
Lock Interval	Time span (in minutes) during which consecutive failed login attempts are counted in determining whether to lock a user account.

Table 7 Password Policy Tab — Field Descriptions (Continued)

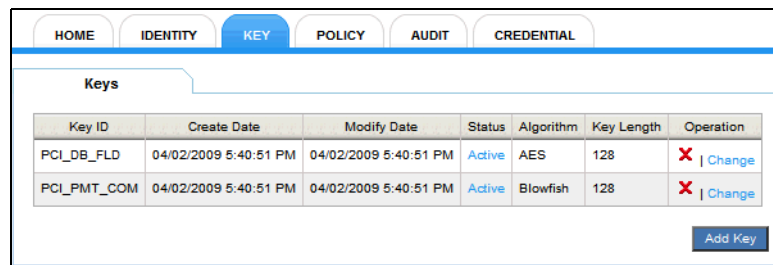
Field	Description
Dictionary List	(No default values.) Dictionary list, which is a list of words that are prohibited as passwords. Separate the words with commas. During evaluation of a submitted password, the dictionary list is checked first. If an exact match is found in the dictionary list, no further checking of password constraints occurs and the password cannot be used. If a match is not found in the dictionary list, then the password is evaluated against other parts of the password policy.
IsDefaultPolicy	Indicates whether the password policy being used is the default password policy (that is, whether values in all fields on the Password Policy tab are from the default password policy).



Key Management

This section discusses symmetric encryption keys, which are centrally managed at the Security Server. For background information, see [“Encryption Key and Credentials Management” on page 61](#).

Viewing Keys

To view the currently defined symmetric encryption keys, select the top-level **Key** tab on any page in the Security GUI. A page appears with a single Keys tab that shows the existing keys. [Figure 75](#) shows an example of the tab.

Figure 75 Security GUI — Keys Tab


Keys						
Key ID	Create Date	Modify Date	Status	Algorithm	Key Length	Operation
PCI_DB_FLD	04/02/2009 5:40:51 PM	04/02/2009 5:40:51 PM	Active	AES	128	 Change
PCI_PMT_COM	04/02/2009 5:40:51 PM	04/02/2009 5:40:51 PM	Active	Blowfish	128	 Change

[Add Key](#)

Creating Keys

To create a new symmetric key in the Security Server database, do the following:

1. Select the top-level **Key** tab on any page in the Security GUI.
A page appears with a single Keys tab that shows the existing keys (see [Figure 75](#)).
2. On the Keys tab, click **Add Key**.
A new page appears that contains the Add Key Details tab where you can define information for the key. [Figure 76 on page 100](#) shows the tab.

Figure 76 Security GUI — Add Key Details Tab

3. Fill in the fields on the Add Key Details tab (red asterisks indicate required fields). [Table 8](#) describes the fields.

Table 8 Add Key Details Tab — Field Descriptions

Field	Description
Key ID	Leave this field blank if you want the system to create the key ID. Otherwise, enter a well-known custom name that one or more applications use to look up the key. This custom name can contain letters, digits, and special characters (such as underscore and hyphen).
Algorithm	Select the algorithm for which the key is being created (AES or Blowfish).

4. After filling in the fields, click **Save**.
A message appears on the screen indicating that the operation was successful.

Disabling/Enabling Keys

Occasionally, you might need to disable a symmetric key. Disabling a symmetric key changes its status from active to inactive, meaning it is no longer usable. After a key has been disabled, it might be necessary to enable (activate) it again.

To disable/enable a key, do the following steps:

1. Select the top-level **Key** tab on any page in the Security GUI.
A page appears with a single Keys tab that shows the existing keys (see [Figure 75 on page 99](#)).
2. On the Keys tab, click the value in the Status column.
Clicking the value toggles the key status from active to inactive, and vice versa. The value in the Status field changes accordingly, and a message appears on the screen indicating that the operation was successful.

Deleting Keys

In any situation where a key is no longer used, the old key should be deleted from the Security Server database. Before a symmetric key can be deleted, its status must be inactive.

To delete a key, do the following steps:

1. If the key's status is active, change it to inactive. (See [“Disabling/Enabling Keys”](#) for information on changing the status to inactive.)
2. On the Keys tab (see [Figure 75 on page 99](#)), click the Delete (red X) button in the Operation column for the key you want to delete.
A message appears on the screen indicating that the operation was successful.

Policy Management

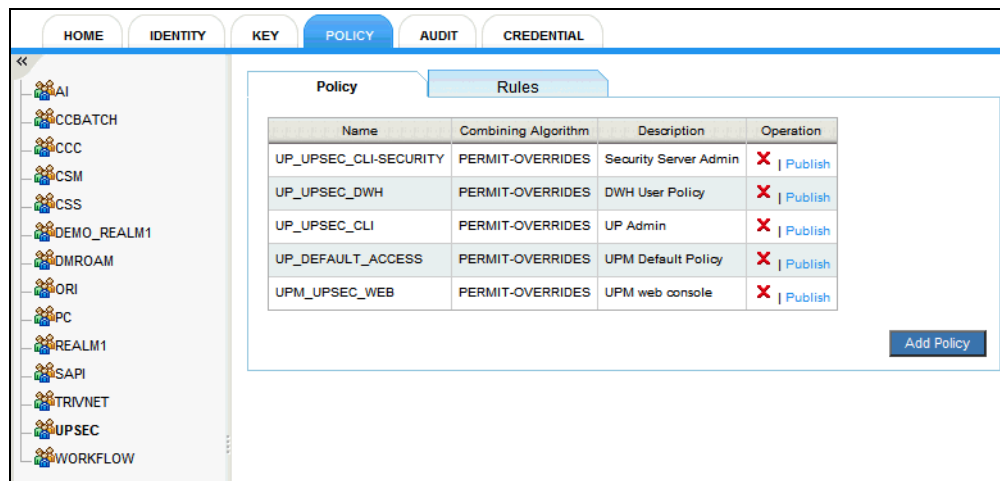
Policy management operations enable you to work with the following:

- **Authorization Policies:** Policies are defined per security realm.
- **Authorization Rules:** Rules are independent of security realms.

For background information, see [“Policy Management Overview” on page 35](#).

To go to the Policy Management area, select the top-level **Policy** tab on any page in the Security GUI. The Policy Management page appears. [Figure 77](#) shows an example of the page, with the UPSEC security realm (the realm for administrators) selected by default. The Policy tab lists all policies defined for that realm.

Figure 77 Security GUI — Policy Tab



On the left side of the Policy Management page is a navigation tree of currently defined security realms. Policies are defined per security realm, so any work done on policies is for the selected security realm.

Working with Policies

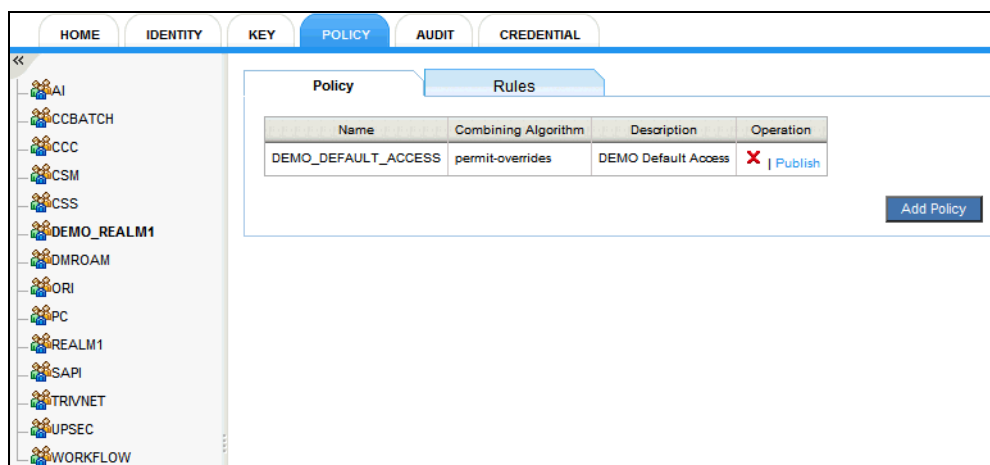
The following sections provide information on working with authorization policies. In practical terms, a policy is a set of rules defining which subjects (currently, roles) are permitted to access which resources using which actions. Policies are defined per security realm.

Viewing Policies

To view policies defined for a security realm, do the following steps:

1. Select the top-level **Policy** tab on any page in the Security GUI.
2. On the Policy Management page that appears, select the appropriate security realm from the realm navigation tree.

The Policy tab lists all currently defined policies for the selected realm. [Figure 78 on page 102](#) shows an example.

Figure 78 Security GUI — Policy Tab Showing Policies for a Realm

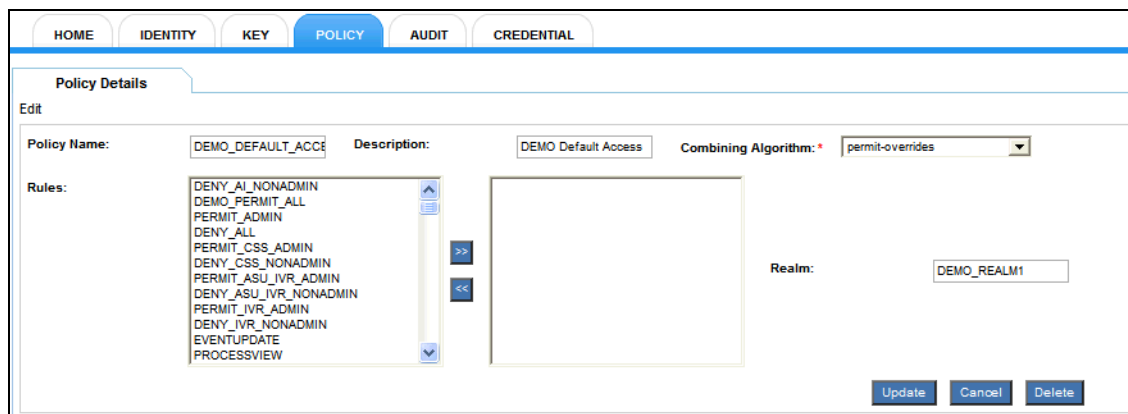
From the Policy tab, you can do the following operations on a policy:

- **View/Modify All Data for a Policy:** To view /modify all data for the policy, click the row for the policy. (For details on modifications, see [“Modifying Policies”](#)).
- **Delete a Policy:** Click the Delete (red X) button in the Operation column for the policy you want to delete. For more information, see [“Deleting Policies” on page 103](#).
- **Publish a Policy:** For information, see [“Publishing/Resynchronizing Policies” on page 105](#).
- **Add a New Policy:** For information, see [“Creating Policies” on page 103](#).

Modifying Policies

To modify a policy, do the following steps:

1. View policies for the appropriate security realm by following instructions in [“Viewing Policies” on page 101](#).
2. Click the row for the policy you want to modify. A new page appears that contains the Edit Policy Details tab. [Figure 79](#) shows an example.

Figure 79 Security GUI — Edit Policy Details Tab (Modifying a Policy)

3. Make any needed modifications on the Edit Policy Details tab.
For descriptions of fields on the tab, see [Table 9, “Policy Details Tab — Field Descriptions,” on page 104](#). Although that table is for adding policy details, it contains field descriptions relevant to editing policy details.

4. After finishing your modifications, click **Update**.

A message appears on the screen indicating that the operation was successful.

**NOTE**

Modifications to a policy only affect the database representation of the policy and do not affect the currently published policy being used by an application. The policy must be republished before changes take effect for the target application. See [“Publishing/Resynchronizing Policies” on page 105](#). Republishing a policy is necessary only for such modifications as changes to rules or the rule-combining algorithm. A change to the policy description does not require republishing.

Deleting Policies

Two methods are available to delete policies.

For the first method, do the following steps:

1. View policies as described in [“Viewing Policies” on page 101](#).
2. On the Policy tab that lists the current policies for the selected security realm (see [Figure 78 on page 102](#) for an example), click the Delete (red X) button in the Operation column for the policy that you want to delete.

A dialog box appears, asking you to confirm the deletion.

3. Click **OK** in the dialog box.

A message appears on the screen indicating that the operation was successful.

For the second method, do the following steps:

1. View policies as described in [“Viewing Policies” on page 101](#).
2. On the Policy tab that lists the policies defined for the selected security realm (see [Figure 78 on page 102](#) for an example), click the row for the policy.
3. On the Edit Policy Details tab that is displayed (see [Figure 79 on page 102](#)), click **Delete**.

A message appears on the screen indicating that the operation was successful.

Deleting a policy removes it from the Security Server database. If the policy has been published, you must publish and resynchronize policies for the target node(s) again to remove the policy from the target node(s). See [“Publishing/Resynchronizing Policies” on page 105](#).

Creating Policies

Policies are defined per security realm.

**NOTE**

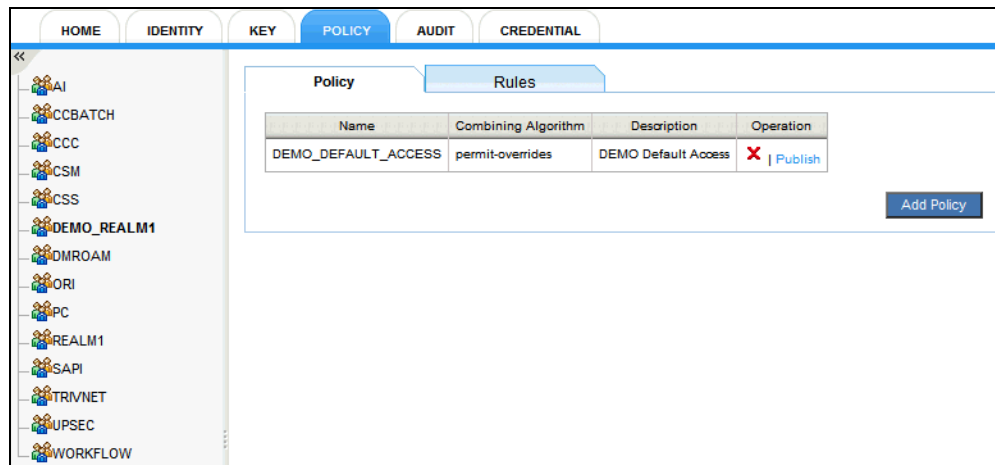
Rules that you will use in the policy must already exist. See [“Creating Rules” on page 108](#).

To create a policy, do the following steps:

1. Select the top-level **Policy** tab on any page in the Security GUI.
2. On the Policy Management page that appears, select the appropriate security realm from the realm navigation tree.

3. Do one of the following:
 - a. On the Policy Management page, select the Policy tab. This tab lists the policies currently defined for the selected security realm. [Figure 80](#) shows an example. Click **Add Policy** on the Policy tab.

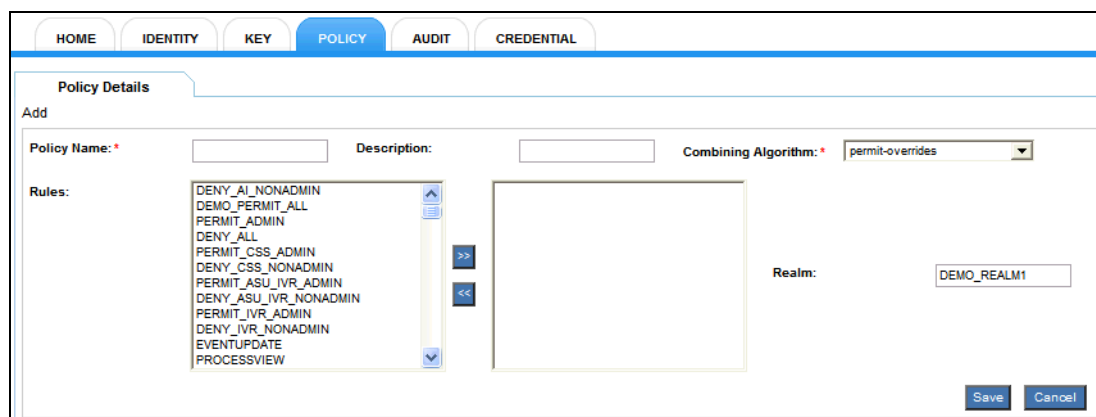
Figure 80 Security GUI — Policy Tab (Creating a Policy)



- b. As an alternative to step 3a, right-click the realm name in the realm navigation tree and select **Add Policy** from the popup menu.

After you do step 3a or 3b, a new page appears with an Add Policy Details tab where you can define the policy. [Figure 81](#) shows the tab.

Figure 81 Security GUI — Add Policy Details Tab (Creating a Policy)



4. Fill in the fields on the Add Policy Details tab (red asterisks indicate required fields). [Table 9](#) describes the fields.

Table 9 Policy Details Tab — Field Descriptions

Field	Description
Policy Name	Name of the policy. Identify the security realm to which the policy applies by using the realm name as the first few characters of the policy name. Two examples of policy names are PC_<SomeName> and SAPI_<SomeName>, where PC and SAPI are realms for the Product Catalog and Single API. Activities such as publishing and resynchronizing policies rely on this naming convention.

Table 9 Policy Details Tab — Field Descriptions (Continued)

Field	Description
Description	Description of the policy.
Combining Algorithm	Rule-combining algorithm to resolve any conflicts among rules in the policy. Valid values are the following: permit-overrides — If any rule evaluates to Permit, the final authorization decision is also Permit. deny-overrides — If any rule evaluates to Deny, the final authorization decision is also Deny. first-applicable — The result of the first relevant rule encountered is the final authorization decision.
Rules	The list on the left shows all the currently defined rules. The list on the right shows the rules currently associated with the policy. Click the >> and << icons to move the selected rule(s) from one list to the other.
Realm	Security realm for which the policy is being defined. (This field is auto-filled by the selection from the realm navigation tree.)

- After filling in fields on the Add Policy Details tab, click **Save**.

A message appears on the screen indicating that the operation was successful. The tab changes from Add Policy Details to Edit Policy Details to enable modifications.

Publishing/Resynchronizing Policies



NOTE

Activities such as publishing policies and resynchronizing policies rely on a policy-naming convention. The first few characters of the policy name must be a realm name that identifies the realm to which the policy applies. Two examples of policy names are PC_<SomeName> and SAPI_<SomeName>, where PC and SAPI are realms for the Product Catalog and Single API.

After a policy is created or modified, it must be published and resynchronized with target nodes in order to take effect for target applications. For background information, see [“Publishing Policies” on page 43](#).

To publish/resynchronize a policy, do the following steps:

- Select the top-level **Policy** tab on any page in the Security GUI.
- On the Policy Management page that appears, select the appropriate security realm from the navigation tree.
 Policies currently defined for the selected security realm are displayed on the Policy tab (see [Figure 80 on page 104](#) for an example).
- Click the **Publish** link in the Operation column for the policy you want to publish.
 A new page appears containing a Publish Policy Details tab, with the selected policy name filled in by default. [Figure 82 on page 106](#) shows an example of the tab.

Figure 82 Security GUI — Publish Policy Details Tab

4. Fill in data for the Node Class or Node Name field on the Publish Policy Details tab.
5. Click **Publish**.

A message appears on the screen indicating that the operation was successful, and the tab is updated with results of the publish/resynchronize operation. [Figure 83](#) shows an example.

Figure 83 Security GUI — Publish Policy Details Tab (Publish/Resynchronize Results)

Policy Id	Status	NodeClass	NodeName	Node IP
UP_DEFAULT_ACCESS	Publish Success	MANAGER	--	10.210.156.167
UP_DEFAULT_ACCESS*	Resync Success - 1	MANAGER	upm1	10.210.156.167

Working with Rules

The following sections provide information on working with authorization rules. A rule, which is independent of security realms, is the most elementary unit of an authorization policy.



NOTE

There is no one-to-one mapping of rule to policy. That is, a rule is independent of any given policy, and the same rule can be used in multiple policies. Rules are independent of security realms.

Viewing Rules

To view currently defined authorization rules, do the following steps:

1. Select the top-level **Policy** tab on any page in the Security GUI.
2. On the Policy Management page that appears, select the Rules tab.

The Rules tab lists all currently defined rules. [Figure 84 on page 107](#) shows an example of the tab.

Figure 84 Security GUI — Rules Tab Showing Current Rules

Policy	Rules							
	Rule Id	Subject	Resource	Action	Effect	CreateDate	Description	Operation
	PC_ACTION_SELECT_SERVICE_VERSION1	RESELLERALLLAYER,RESELLERALLLAYERANDPF	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Select service version	✖
	PC_ACTION_PROPAGATE_PC_VERSION1	RESELLERALLLAYER,RESELLERMARKETINGLAYER	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Propagate PC version	✖
	PC_ACTION_EDIT_TARGET_DATABASE1	RESELLERALLLAYER,RESELLERALLLAYERANDPF	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Edit target database	✖
	PC_ACTION_COMPARE_VERSIONS	OPERATORSYSTEMDEPLOYMENTADMIN,OPERAT	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Compare version	✖
	PC_ACTION_EDIT_TARGET_GROUP	OPERATORSYSTEMDEPLOYMENTADMIN,OPERAT	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Edit target group	✖
	PC_ACTION_MAKE_CORRECTIVE_CHANGES1	RESELLERALLLAYER,RESELLERMARKETINGLAYER	COM.COMVERSE\MODULES\UPCLUI\WIZARD\CI	ANY	DENY	12/27/2008 10:12:58 AM	Make corrective version	✖
	PC_ACTION_PERMIT_ALL_OPERATOR	OPERATORSYSTEMDEPLOYMENTADMIN,OPERAT	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	PERMIT	12/27/2008 10:12:58 AM	Catch all rule for all actions	✖
	PC_ACTION_MAKE_CORRECTIVE_CHANGES	OPERATORSYSTEMDEPLOYMENTADMIN,OPERAT	COM.COMVERSE\MODULES\UPCLUI\WIZARD\CI	ANY	DENY	12/27/2008 10:12:58 AM	Make corrective version	✖
	PC_ACTION_CREATE_TARGET_DATABASE1	RESELLERALLLAYER,RESELLERALLLAYERANDPF	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Create target database	✖
	PC_ACTION_CHECKOUT_SERVICE_VERSION1	RESELLERALLLAYER,RESELLERALLLAYERANDPF	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Check out service version	✖
	PC_ACTION_EDIT_REGION	OPERATORRESELLERADMIN,OPERATORBASICA	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Edit region	✖
	PC_ACTION_PROMOTE_PC_VERSION	OPERATORSYSTEMDEPLOYMENTADMIN,OPERAT	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Promote PC version	✖
	PC_ACTION_CHECKOUT_SERVICE_VERSION	OPERATORSYSTEMDEPLOYMENTADMIN,OPERAT	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Check out service version	✖
	PC_ACTION_CORRECT_PC_VERSION1	RESELLERALLLAYER,RESELLERMARKETINGLAYER	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Correct PC version	✖
	PC_ENTITY_CALLING_CIRCLE_EDIT	RESELLERSERVICEAYERCCVIEWONLY	COM.COMVERSE\PCFWA\ENTITY\CALLINGCIRC	EDIT	DENY	12/27/2008 10:12:58 AM	Edit Calling Circle entities in service layer	✖
	PC_ACTION_VIEW_EDIT_PC_VERSION1	RESELLERALLLAYER,RESELLERMARKETINGLAYER	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Edit PC version	✖
	PC_ACTION_CREATE_REGION	OPERATORRESELLERADMIN,OPERATORBASICA	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Create new region	✖
	PC_ACTION_EDIT_TARGET_GROUP1	RESELLERALLLAYER,RESELLERALLLAYERANDPF	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Edit target group	✖
	PC_ACTION_VIEW_EDIT_PC_VERSION	OPERATORSYSTEMDEPLOYMENTADMIN,OPERAT	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Edit PC version	✖
	PC_ACTION_REJECT_PC_VERSION	OPERATORSYSTEMDEPLOYMENTADMIN,OPERAT	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Reject PC version	✖

148 records found, displaying 20 records, from 1 to 20. Page 1 / 8.

1234

Add Rule

Modifying Rules

To modify a rule, do the following steps:

1. View existing rules as described in [“Viewing Rules” on page 106](#).
2. On the Rules tab (shown in [Figure 84](#)), click the row for the rule you want to modify. A new page appears containing the Edit Rule tab. [Figure 85](#) shows an example.

Figure 85 Security GUI — Edit Rule Tab (Modifying a Rule)

HOME
IDENTITY
KEY
POLICY
AUDIT
CREDENTIAL

Edit Rule

Rule Id: * DEMO_PERMIT_ALL
Subject: * Any
Resource: * Any
Action: * Any
Effect: * Permit
Description: Demo Permit Everything

Update Rule
Delete
Cancel

3. Make all needed modifications on the Edit Rule tab. For descriptions of fields on the tab, see [Table 10, “Add Rule Tab — Field Descriptions,” on page 109](#). Although that table is for adding rules, it contains field descriptions relevant to modifying rules.
4. Click **Update Rule**.

A message appears on the screen indicating that the operation was successful.

**NOTE**

Modification of an authorization rule does not affect published authorization policies that currently use the rule. (Applications use policies only after they have been published.) Policies must be republished before changes to their rules take effect for target applications. See [“Publishing/Resynchronizing Policies” on page 105](#). Note that changing a rule description does not require republishing the policy.

Deleting Rules

Two methods are available to delete an authorization rule.

For the first method, do the following steps:

1. View rules as described in [“Viewing Rules” on page 106](#).
2. On the Rules tab that lists all defined rules, click the Delete (red X) button in the Operation column for the rule you want to delete.
A dialog box appears, asking you to confirm the deletion.
3. Click **OK** in the dialog box.
A message appears on the screen indicating that the operation was successful.

For the second method, do the following steps:

1. View rules as described in [“Viewing Rules” on page 106](#).
2. On the Rules tab that lists all defined rules, click the row for the rule you want to delete.
3. On the Edit Rule tab that is displayed (see [Figure 85 on page 107](#) for an example), click **Delete**.
A message appears on the screen indicating that the operation was successful.

**NOTE**

Deletion of a rule does not affect published policies that currently use the rule. Policies must be republished before a rule deletion takes effect. See [“Publishing/Resynchronizing Policies” on page 105](#).

Creating Rules

Authorization rules are independent of security realms.

To create a new authorization rule, do the following steps:

1. Select the top-level **Policy** tab on any page in the Security GUI.
2. On the Policy Management page that appears, select the Rules tab.
The Rules tab lists all currently defined rules. [Figure 86 on page 109](#) shows an example of the tab.

Figure 86 Security GUI — Rules Tab (Creating a Rule)

Policy		Rules						
Rule Id	Subject	Resource	Action	Effect	CreateDate	Description	Operation	
PC_ACTION_SELECT_SERVICE_VERSION1	RESELLERALLLAYER,RESELLERALLLAYERANDPF	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Select service version	✗	
PC_ACTION_PROPAGATE_PC_VERSION1	RESELLERALLLAYER,RESELLERMARKETINGLAYER	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Propagate PC version	✗	
PC_ACTION_EDIT_TARGET_DATABASE1	RESELLERALLLAYER,RESELLERALLLAYERANDPF	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Edit target database	✗	
PC_ACTION_COMPARE_VERSIONS	OPERATORSYSTEMDEPLOYMENTADMIN,OPERAT	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Compare version	✗	
PC_ACTION_EDIT_TARGET_GROUP	OPERATORSYSTEMDEPLOYMENTADMIN,OPERAT	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Edit target group	✗	
PC_ACTION_MAKE_CORRECTIVE_CHANGES1	RESELLERALLLAYER,RESELLERMARKETINGLAYER	COM.COMVERSE\MODULES\UPCLUI\WIZARD\CI	ANY	DENY	12/27/2008 10:12:58 AM	Make corrective version	✗	
PC_ACTION_PERMIT_ALL_OPERATOR	OPERATORSYSTEMDEPLOYMENTADMIN,OPERAT	COM.COMVERSE\MODULES\UPCLUI*	ANY	PERMIT	12/27/2008 10:12:58 AM	Catch all rule for all actions	✗	
PC_ACTION_MAKE_CORRECTIVE_CHANGES	OPERATORSYSTEMDEPLOYMENTADMIN,OPERAT	COM.COMVERSE\MODULES\UPCLUI\WIZARD\CI	ANY	DENY	12/27/2008 10:12:58 AM	Make corrective version	✗	
PC_ACTION_CREATE_TARGET_DATABASE1	RESELLERALLLAYER,RESELLERALLLAYERANDPF	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Create target database	✗	
PC_ACTION_CHECKOUT_SERVICE_VERSION1	RESELLERALLLAYER,RESELLERALLLAYERANDPF	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Check out service version	✗	
PC_ACTION_EDIT_REGION	OPERATORRESELLERADMIN,OPERATORBASICLA	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Edit region	✗	
PC_ACTION_PROMOTE_PC_VERSION	OPERATORSYSTEMDEPLOYMENTADMIN,OPERAT	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Promote PC version	✗	
PC_ACTION_CHECKOUT_SERVICE_VERSION	OPERATORSYSTEMDEPLOYMENTADMIN,OPERAT	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Check out service version	✗	
PC_ACTION_CORRECT_PC_VERSION1	RESELLERALLLAYER,RESELLERMARKETINGLAYER	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Correct PC version	✗	
PC_ENTITY_CALLING_CIRCLE_EDIT	RESELLERSERVICEAYERCCVIEWONLY	COM.COMVERSE\PCFWA\ENTITY\CALLINGCIRC	EDIT	DENY	12/27/2008 10:12:58 AM	Edit Calling Circle entities in service layer	✗	
PC_ACTION_VIEW_EDIT_PC_VERSION1	RESELLERALLLAYER,RESELLERMARKETINGLAYER	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Edit PC version	✗	
PC_ACTION_CREATE_REGION	OPERATORRESELLERADMIN,OPERATORBASICLA	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Create new region	✗	
PC_ACTION_EDIT_TARGET_GROUP1	RESELLERALLLAYER,RESELLERALLLAYERANDPF	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Edit target group	✗	
PC_ACTION_VIEW_EDIT_PC_VERSION	OPERATORSYSTEMDEPLOYMENTADMIN,OPERAT	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Edit PC version	✗	
PC_ACTION_REJECT_PC_VERSION	OPERATORSYSTEMDEPLOYMENTADMIN,OPERAT	COM.COMVERSE\MODULES\UPCLUI\ACTIONS\I	ANY	DENY	12/27/2008 10:12:58 AM	Reject PC version	✗	

148 records found, displaying 20 records, from 1 to 20. Page 1 / 8.

- On the Rules tab, click **Add Rule**.

A new page appears containing the Add Rule tab. [Figure 87](#) shows an example.

Figure 87 Security GUI — Add Rule Tab (Creating a Rule)

HOME	IDENTITY	KEY	POLICY	AUDIT	CREDENTIAL
Add Rule					
Rule Id: *	<input type="text"/>		Subject: *	<input type="text"/>	
Resource: *	<input type="text"/>		Action: *	<input type="text"/>	
Effect: *	<input type="text" value="-Select One-"/>		Description:	<input type="text"/>	
				Save Rule	Cancel

- Fill in the fields on the Add Rule tab (red asterisks indicate required fields). [Table 10](#) describes the fields.

Table 10 Add Rule Tab — Field Descriptions

Field	Description
Rule ID	Unique name for the rule.
Subject	Subject of the rule, currently a role. (Although subjects can be many things, currently Comverse ONE applications use roles as the subjects of rules.) When providing multiple subjects as values, separate the subjects with commas. You can specify a value of ANY to include all subjects.
Resource	Resource of the rule. Resources are application-specific and can be data, service, or system components. For example, service methods can be resources. When providing multiple resources as values, separate the resources with commas. You can specify a value of ANY to include all resources.

Table 10 Add Rule Tab — Field Descriptions (Continued)

Field	Description
Action	Action of the rule. An action is an operation on a resource. The types of actions are application-specific. Some examples are invoke, enable, disable, read, write, create, update, delete, and so on. When providing multiple actions as values, separate the actions with commas. You can specify a value of ANY to include all actions.
Effect	Intended consequence of the satisfied rule. From the dropdown list, select either Permit or Deny.
Description	Description of the rule.

- After filling in fields on the Add Rule tab, click **Save Rule**.

A message appears on the screen indicating that the operation was successful.

Audit Management

The Security GUI provides the ability to view audit records. For background information, see [“Audit Management” on page 47](#).

To view audit records, do the following steps:

- Select the top-level **Audit** tab on any page in the Security GUI.

A page appears with a single Audit Records tab that shows the records. By default, records from the start of the current day (midnight) to the current time are displayed. Records are sorted by Time Offset, in ascending order. [Figure 88](#) shows an example of the Audit Records tab.

Figure 88 Security GUI — Audit Records Tab

Time offset	UserName	Event Outcome	Event Number	Originator Address	Originator Name	Target Principal Name	Event Info
2009-05-28 08:35:32.0	--	0	16777251	10.210.156.160	devsite/upsec/upm1/manager	getRealm	realm=UPSEC,message=Retrieving Realm Success
2009-05-28 08:35:33.0	secadmin	0	16777223	10.210.156.160	devsite/upsec/upm1/manager	Login	command=Login,method=null
2009-05-28 08:35:40.0	--	0	16777251	10.210.156.160	devsite/upsec/upm1/manager	listUsers	realm=UPSEC,message=Retrieving List of Users Success
2009-05-28 08:35:41.0	--	0	16777251	10.210.156.160	devsite/upsec/upm1/manager	listRealms	message=Retrieving List of Realms Success
2009-05-28 08:35:42.0	--	0	16777251	10.210.156.160	devsite/upsec/upm1/manager	listUsers	realm=UPSEC,message=Retrieving List of Users Success
2009-05-28 08:40:46.0	secadmin	1026	16777223	10.210.156.160	devsite/upsec/upm1/manager	Login	command=Login,method=null
2009-05-28 08:40:58.0	--	0	16777251	10.210.156.160	devsite/upsec/upm1/manager	getRealm	realm=UPSEC,message=Retrieving Realm Success
2009-05-28 08:40:58.0	secadmin	0	16777223	10.210.156.160	devsite/upsec/upm1/manager	Login	command=Login,method=null
2009-05-28 08:43:13.0	--	0	16777251	10.210.156.160	devsite/upsec/upm1/manager	getRealm	realm=UPSEC,message=Retrieving Realm Success
2009-05-28 08:43:15.0	secadmin	0	16777223	10.210.156.160	devsite/upsec/upm1/manager	Login	command=Login,method=null

In [Figure 88](#), the values in the Event Outcome and Event Number columns are the decimal representations of the hex XDAS codes discussed in [“XDAS Event Outcome Codes” on page 57](#) and [“XDAS Event Codes” on page 53](#).

2. To filter the list of audit records on the Audit Records tab, provide data in the fields as follows:
 - ❑ **UserID:** Records for a specific user.
 - ❑ **CommandName:** Records corresponding to a specific command (for example, login).
 - ❑ **ExternalID:** Records corresponding to an external ID, such as a subscriber number.
 - ❑ **From Date/To Date:** Records from the date specified to the date specified.
3. Click **Filter** to filter the records based on the data you provided.
The display of records changes accordingly.

Credentials Management

The term “credentials” refers to passwords for business databases used by Comverse ONE applications and SNMP community strings for network devices.

During upgrades, database passwords are reverted to their default passwords. After an upgrade, you can change the passwords again. To do this, delete the passwords as discussed in [“Deleting Database Credentials” on page 112](#), create the new passwords as discussed in [“Creating Database Credentials” on page 112](#), and publish the passwords (and also publish to cache) as discussed in [“Publishing Database Credentials” on page 113](#).

Viewing Database Credentials

To view database credentials (database passwords), select the top-level **Credential** tab on any page in the Security GUI. The Credentials page appears, with the Database tab selected by default. The Database tab lists all currently defined database credentials. [Figure 89](#) shows an example.

Figure 89 Security GUI — Credentials, Database Tab

HOME IDENTITY KEY POLICY AUDIT CREDENTIAL					
Credentials					
Database Network					
User ID	DB Type	Instance	Password	Operation	
UPMUSER	UPM	XE	comverse	✗	Publish
SECUSER	UPM	XE	comverse	✗	Publish
WPUSER	UPM	XE	comverse	✗	Publish
SYSTEM	UPM	XE	mg518	✗	Publish
SYSTEM	RATING	MAIN	mg518	✗	Publish
CBS_OWNER	CBS	MAIN	comverse	✗	Publish
CBS_OWNER	CBS	HIST	comverse	✗	Publish
CBS_OWNER	CBS	BLUS	comverse	✗	Publish
CBS_OWNER	CBS	CTLG1	comverse	✗	Publish
CBS_OWNER	CBS	CTLG	comverse	✗	Publish
CBS_OWNER	PCAT	MAIN	comverse	✗	Publish
CBS_OWNER	RATING	MAIN	comverse	✗	Publish
CBS_OWNER	BILLING	CUST1	comverse	✗	Publish
CBS_OWNER	UPC	MAIN	comverse	✗	Publish
CBS_OWNER	UPC	PCAT	comverse	✗	Publish
DROP31_TR8_DEV	RATING	MAIN	arbor123	✗	Publish
Add Database Credential					

Deleting Database Credentials

To delete a database credential (database password), do the following steps:

1. View database credentials as described in [“Viewing Database Credentials.”](#)
2. On the Database tab (see [Figure 89 on page 111](#)), click the Delete (red X) button in the Operation column.
A dialog box appears, asking you to confirm the deletion.
3. Click **OK** in the dialog box.
A message appears on the screen indicating that the operation was successful.



NOTE

To change an existing database password, the database credential is first deleted as described here and then a new one is created (see [“Creating Database Credentials.”](#)). If a database password has already been published (that is, propagated to the target database), it remains in effect for the target database until credentials are published again.

Creating Database Credentials

To create a database credential (database password), do the following steps:

1. Select the top-level **Credential** tab on any page in the Security GUI.
The Credentials page appears, with the Database tab selected by default (see [Figure 89 on page 111](#)).
2. On the Database tab that shows the currently defined database credentials, click **Add Database Credential**.
A new page appears, containing the Add Credential Details tab. [Figure 90](#) shows the tab.

Figure 90 Security GUI — Add Credential Details Tab (Creating Database Credentials)

3. Fill in fields on the Add Credential Details tab (red asterisks indicate required fields). [Table 11](#) describes the fields.

Table 11 Add Credential Details Tab — Field Descriptions

Field	Description
User ID	Unique database user ID.
DB Type	Database type (such as RATING, PCAT, CBS and so on).
Instance	Database instance name (such as MAIN or HIST if the database type is RATING).
Password	Password for the database user ID.

4. After filling in fields on the Add Credential Details tab, click **Save Credential**.

A message appears on the screen indicating that the operation was successful.



NOTE

After creation, a database password exists in the Security Server database. It must be propagated to the target database (that is, published) to activate it for use. For information, see [“Publishing Database Credentials”](#).

Publishing Database Credentials



NOTE

Publishing database credentials should be done during a maintenance window (that is, when the node where the database resides is in maintenance mode).

To publish a database credential (database password), do the following steps:

1. View database credentials as described in [“Viewing Database Credentials” on page 111](#).
2. On the Database tab that shows currently defined database credentials (see [Figure 89 on page 111](#)), click the **Publish** link in the Operation column.

A new page appears, containing the Publish Credential Details tab. [Figure 91](#) shows an example of the tab.

Figure 91 Security GUI — Publish Credential Details Tab

3. Fill in fields on the Publish Credential Details tab (red asterisks indicate required fields). [Table 12](#) describes the fields.

Table 12 Publish Credential Details Tab — Field Descriptions

Field	Description
Publish to Cache	Selecting Yes publishes the password to the cache (that is, it refreshes the cache) for use by client applications. Selecting No means the cache is not refreshed.
Node Class	Node class (node type) where the database resides, such as SDP. The password for the given user ID will be propagated to the target database instance on nodes of the given type.
Node Name	Node name where the database resides. The password will be propagated to the target database instance on the given named node.

Table 12 Publish Credential Details Tab — Field Descriptions (Continued)

Field	Description
Node Instance	IP address of the node where the database resides. The password will be propagated to the target database instance on the given node instance.

- After filling in fields on the Publish Credential Details tab, click **Publish Credential**.
A message appears on the screen indicating that the operation was successful.

Viewing Network Credentials

To view network credentials (SNMP community strings for network devices), do the following steps:

- Select the top-level **Credential** tab on any page in the Security GUI.
The Credentials page appears, with the Database tab selected by default.
- Select the Network tab to view all currently defined network credentials. [Figure 92](#) shows an example of the tab.

Figure 92 Security GUI — Credentials, Network Tab

Credentials				
Database Network				
Node Name	Node Class	Instance	SNMP Community String	Operation
cajun	cajun	0.0.0.0	public	X
ciscohub	ciscohub	0.0.0.0	public	X
ciscorouter	ciscorouter	0.0.0.0	public	X
lba	lba	0.0.0.0	public	X
nortel	nortel	0.0.0.0	public	X

Add Network Credential

Deleting Network Credentials

To delete a network credential (SNMP community string), do the following steps:

- View network credentials as described in [“Viewing Network Credentials” on page 114](#).
- On the Network tab (see [Figure 92](#)), click the Delete (red X) button in the Operation column for the network credential you want to delete.
A dialog box appears, asking you to confirm the deletion.
- Click **OK** in the dialog box.
A message appears on the screen indicating that the operation was successful.

Creating Network Credentials

To create a network credential (SNMP community string for a network device), do the following steps:

- Select the top-level **Credential** tab on any page in the Security GUI.
The Credentials page appears, with the Database tab selected by default.
- Select the Network tab (see [Figure 92](#)) to view all currently defined network credentials.

3. On the Network tab, click **Add Network Credential**.

A new page appears containing the Add Network Credential tab where you can define the new network credential. [Figure 93](#) shows the tab.

Figure 93 Security GUI — Add Network Credential Tab

4. Fill in fields on the Add Network Credential tab (red asterisks indicate required fields). [Table 13](#) describes the fields.

Table 13 Add Network Credential Tab — Field Descriptions

Field	Description
Node Class	Node class of the network device.
Node Name	Node name of the network device.
Instance	Node instance (IP address) of the network device.
SNMP Community String	SNMP community string for the network device.

5. After filling in fields on the Add Network Credential tab, click **Save**.
A message appears on the screen indicating that the operation was successful.

Appendix A

Security-Related Management Shell Commands

A

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

v

Overview

For your convenience, this appendix describes all security-related commands that can be executed using the Management Shell (mshell) command line interface (CLI). The discussion for each command includes all its command-specific options, with information on which options are mandatory.

Commands in this appendix are grouped as follows:

- [“Identity Management Commands” on page 120](#)
- [“Policy Management Commands” on page 129](#)
- [“Audit Management Commands” on page 133](#)
- [“Key Management Commands” on page 134](#)
- [“Credentials Management Commands” on page 135](#)

Within each group, commands appear in alphabetical order.

Global Options

In addition to the command-specific options described for each command, the following global options can also be used:

- i Ignore fields when displaying data. The term “fields” refers to comma-separated column names to exclude from the displayed output.
- f Filter records when displaying data. The format for input to this option is “<ColumnName>:<Value>”, where <Value> can be a literal value or a valid regular expression.
- q Name used for the query type or favorite. For details on this option, see information on the `build_favorite` command in the *Unified Platform Guide*.

General Command Syntax

The general syntax for commands is as follows:

```
command_name <-command_specific_option> <value> <-command_specific_option> <value> <-global_option> <value>
```

The following two examples show the same command with different command-specific options and global options. (See the preceding chapters in this guide for other command examples.)

Example 1:

Display details for the user whose user identifier is `jsmith01` and whose user account is defined in the `DEMO_REALM1` security realm. When showing the details, ignore (exclude from display) the `Department` and `MiddleName` columns.

```
list_users -uid jsmith01 -rlid DEMO_REALM1 -i Department,MiddleName
```

Example 2:

Display a list of users defined in the `DEMO_REALM1` security realm and filter the list to show only those users whose last names begin with the characters “To” followed by any number of any characters (such as `Tocci`, `Tolliver`, `Townes`, and so on).

```
list_users -rlid DEMO_REALM1 -f “LastName:To.*”
```

Identity Management Commands

The following are identity management commands, arranged in alphabetical order.

add_group

Description: The `add_group` command creates a new security realm group associated with the specified security realm.

Command-Specific Options:

- gid Unique group ID (group name). Mandatory option.
- rlid Unique realm ID (realm name) for the group. Mandatory option.
- sdescr Short description of the group.
- desc Description of the group.
- st Soft timeout (maximum period of session inactivity, in minutes) for users in the group, after which a user is automatically logged out and must log in again. The value for soft timeout must be less than the value for hard timeout. If this option is not provided, the value is taken from the default session policy. In that policy, the value for soft timeout is 30 minutes.
- ht Hard timeout (maximum session duration, in minutes) for users in the group, after which a user is automatically logged out and must log in again. The value for hard timeout must be greater than the value for soft timeout. If this option is not provided, the value is taken from the default session policy. In that policy, the value for hard timeout is 480 minutes (8 hours).
- attr Group attributes inherited by users in the group. (When providing multiple attributes as values, separate `<name>: <value>` pairs for attributes with commas. When providing multiple values for an attribute, separate those values with pipe symbols (`|`). Enclose the group of attributes in quotation marks. For example:
-attr "one:1|2|3,two:2".) Be aware that attributes can be defined at the realm, group, and user levels, which means the potential exists for attribute conflicts. See [Appendix B, "Attribute Conflict-Resolution Rules,"](#) for information on how these conflicts are resolved.
- ro Role(s) associated with the group and inherited by all users in the group. (When providing multiple roles as values, separate the roles with commas and enclose the group of roles in quotation marks.)

Restrictions: The security realm must already have been created. If you provide roles, they must already have been created.

add_realm

Description: The `add_realm` command creates a new security realm and, if password options are provided, creates a custom password policy for the realm. (Passwords for all user accounts later associated with the realm must comply with the realm's password policy.) For those password options not specified, values are taken from the default password policy. [Table 4, "Default Password Policy," on page 18](#) describes the default password policy.

Command-Specific Options:

- rlid Unique realm ID (realm name). Mandatory option.
- sdescr Short description of the realm.
- desc Description of the realm.
- attr Realm attributes inherited by all users in the realm, regardless of their group membership. (When providing multiple attributes as values, separate

`<name>: <value>` pairs for attributes with commas. When providing multiple values for an attribute, separate those values with pipe symbols (`|`). Enclose the group of attributes in quotation marks. For example: `-attr "one:1|2|3,two:2"`. Be aware that attributes can be defined at the realm, group, and user levels, which means the potential exists for attribute conflicts. See [Appendix B, "Attribute Conflict-Resolution Rules,"](#) for information on how these conflicts are resolved.

- `-plen` Minimum password length (number of characters). Must be greater than 1.
- `-mxlen` Maximum password length (number of characters). Must be greater than or equal to the minimum length.
- `-ac` Minimum number of alphabetic characters required in a password.
- `-al` Minimum number of the alphabetic characters that must be lowercase.
- `-au` Minimum number of the alphabetic characters that must be uppercase.
- `-oc` Minimum number of other characters (numbers or special characters) required in a password.
- `-md` Minimum number of characters in a new password that must be different from characters in the old password.
- `-mna` Minimum password age, in weeks, before a password can be changed. Mandatory option if `-mxa` is specified.
- `-mxa` Maximum password age, in weeks, at which time the user is notified to change the password. Mandatory option if `-mna` is specified.
- `-mxex` Maximum expiration threshold, which is the number of weeks beyond the maximum age that a password can be changed by the user. After that, it must be reset by the security administrator.
- `-hiex` Number of weeks in which a user cannot reuse a previous password.
- `-hisz` Number of previous passwords that a user cannot reuse.
- `-mxr` Maximum number of consecutive failed login attempts (maximum retries) before a user account is locked.
- `-lkitr` Lock interval, which is the time span (in minutes) during which consecutive failed login attempts are counted in determining whether to lock the user account.
- `-dl` Dictionary list containing comma-separated words that are prohibited as user passwords for the security realm. Enclose the list of words in quotation marks. (During evaluation of a submitted password, the dictionary list is checked first. If an exact match is found in the dictionary list, no further checking of password constraints occurs and the password cannot be used. If a match is not found in the dictionary list, then the password is evaluated against other parts of the password policy.)

Restrictions: None.

add_role

Description: The `add_role` command creates a new security role.

Command-Specific Options:

- `-roid` Unique role ID (role name). Mandatory option.
- `-sdescr` Short description of the role.
- `-descr` Description of the role.

Restrictions: None.

add_user

Description: The `add_user` command creates a new user account associated with the specified security realm.

Command-Specific Options:

- uid Unique user ID. Mandatory option.
- rlid Unique realm ID (realm name) for the user account. Mandatory option.
- fn First name of the user. Mandatory option.
- ln Last name of the user. Mandatory option.
- pwd Password for the user, which must comply with the security realm's password policy. Mandatory option.
- ph Phone number of the user.
- xe Phone extension of the user.
- mn Middle name of the user.
- dept Department of the user.
- gid Group IDs (group names) that the user will belong to. If this option is not provided, the user is placed in the DEFAULT group for the security realm. (When providing multiple groups, separate the groups with commas and enclose the set of groups in quotation marks.)
- attr User attributes, which are custom attributes such as dealer ID and reseller ID. (When providing multiple attributes, separate `<name>: <value>` pairs for attributes with commas. When providing multiple values for an attribute, separate those values with pipe symbols (`|`). Enclose the group of attributes in quotation marks. For example: `-attr "one:1|2|3,two:2"`.) Be aware that attributes can be defined at the realm, group, and user levels, which means the potential exists for attribute conflicts. See [Appendix B, "Attribute Conflict-Resolution Rules,"](#) for information on how these conflicts are resolved.
- lck Lock status for the user account. Valid values are `true` and `false`. If this option is not provided, the default is `false`.
- email Email address of the user.
- acctstate User account state. Valid values are `ENABLED` and `DISABLED`. (Enabled means the account will be purged after a certain number of days of inactivity. Disabled means the account will not be purged because of inactivity. See ["Viewing Purged Inactive User Accounts" on page 28](#) for more details.)
- fcp Force change password at initial login. Valid values are `true` and `false`. If this option is not provided, the default is `false`.
- pgroup Priority group for the user, which serves to resolve attribute conflicts and to establish the soft timeout (session inactivity timeout) and hard timeout (maximum session duration timeout) values for user login sessions. The group must be one of the groups specified for the user with the `-gid` option. If the priority group option is not provided, the user's priority group is the DEFAULT group for the security realm.
- b Batch loading for user accounts and related entities such as security realms, realm groups, and roles. When this option is specified, no other options are recognized. By default, source files that contain user account data and data for the related entities are loaded from the `$JBOSS_HOME/batch/users` directory. An alternate directory can be used. For details, see ["Bulk Account Management Operations" on page 31](#).

Restrictions: Unless the batch load option is used, the security realm and groups (if groups are specified) must already have been created.

change_password

Description: The `change_password` command changes the password of the currently logged-in Management Shell (mshell) user who is executing this command. See [“reset password” on page 129](#) for the command that is used to change passwords for other users.

Command-Specific Options:

- prevpass Old password. Mandatory option.
- newpass New password. Mandatory option.

Restrictions: The new password must comply with the security realm’s password policy.

disable_user

Description: The `disable_user` command disables a user account, which means the account will not be purged because of inactivity. See [“Viewing Purged Inactive User Accounts” on page 28](#) for more details.

Command-Specific Options:

- uid Unique user ID. Mandatory option.
- rlid Unique realm ID (realm name) for the user account. Mandatory option.

Restrictions: None.

enable_user

Description: The `enable_user` command enables a user account, which means the account will be purged after a certain number of days of inactivity. See [“Viewing Purged Inactive User Accounts” on page 28](#) for more details.

Command-Specific Options:

- uid Unique user ID. Mandatory option.
- rlid Unique realm ID (realm name) for the user account. Mandatory option.

Restrictions: None.

find_users

Description: The `find_users` command searches across security realms to find users whose details match the given search criteria.

Command-Specific Options:

- uid User ID for the user.
- fn First name of the user.
- ln Last name of the user.
- lk Lock status of the user account. Valid values are `true` and `false`.
- email Email address of the user.

Restrictions: None.

list_attributes

Description: The `list_attributes` command displays details about the attributes of all users, or a specific user, defined in a security realm. The displayed output shows the origin of the attributes (that is, whether the attributes are defined at the realm, group, or user levels).

Command-Specific Options:

- rlid Unique realm ID. Mandatory option.

-uid User ID of a user.

Restrictions: None.

list_groups

Description: The `list_groups` command provides details for all groups, or an individual group, in the specified security realm.

Command-Specific Options:

-rlid Unique realm ID (realm name) for the group(s). Mandatory option.

-gid Unique group ID (group name) of an individual group whose details you want to view.

Restrictions: None.

list_purged_users

Description: The `list_purged_users` command provides details about user accounts that have been removed (that is, purged) because of inactivity. A purged account is not displayed with the `list_users` command. If a user account is inactive for the configured number of days (the default is 30 days), the user account record is deleted from the `SEC_IDM_USER` table and inserted in the `SEC_IDM_PURGED_USERS` table. A job runs once a day to purge appropriate accounts. Another job runs once a week against the `SEC_IDM_PURGED_USERS` table. When the number of records in that table is greater than the configured allowable number (the default is 50), all records in the table are deleted.

Two configurable system properties specify (1) the number of days a user account can remain inactive before being purged and (2) the number of purged inactive user records allowed in the `SEC_IDM_PURGED_USERS` table:

`user.idletime` = days an account can remain inactive

`idleusers.count` = number of purged inactive records allowed

These properties are located in the `$JBOSS_HOME/conf/application.properties` file.

Command-Specific Options: None.

Restrictions: None.

list_realms

Description: The `list_realms` command provides details for all security realms or for an individual realm. If the `-pp` option is provided, the command shows password policy details for the realm(s) instead of the normally displayed realm details. Included in the details for the `-pp` option is whether or not the password policy being used is the default password policy.

Command-Specific Options:

-rlid Unique realm ID (realm name) of an individual realm whose details you want to view.

-pp Shows password policy details for the security realm(s) instead of the normally displayed realm details.

Restrictions: None.

list_roles

Description: The `list_roles` command provides details for all security roles or for an individual role.

Command-Specific Options:

-roid Unique role ID (role name) of an individual role whose details you want to view.

Restrictions: None.

list_users

Description: The `list_users` command provides details for all user accounts, or for an individual account, in the specified security realm.

Command-Specific Options:

-rlid Unique realm ID (realm name) where the user accounts are defined. Mandatory option.

-uid User ID for an individual user account whose details you want to view.

Restrictions: None.

lock_user

Description: The `lock_user` command locks a user account, disabling logins.

Command-Specific Options:

-uid Unique user ID. Mandatory option.

-rlid Unique realm ID (realm name) for the user account. Mandatory option.

Restrictions: None.

modify_group

Description: The `modify_group` command modifies or adds information for a realm group in the specified security realm.

Command-Specific Options:

-gid Unique group ID (group name). Mandatory option.

-rlid Unique realm ID (realm name) where the group is defined. Mandatory option.

-sdescr Short description of the group.

-desc Description of the group.

-st Soft timeout (maximum period of session inactivity, in minutes) for users in the group, after which a user is automatically logged out and must log in again. The value for soft timeout must be less than the value for hard timeout.

-ht Hard timeout (maximum session duration, in minutes) for users in the group, after which a user is automatically logged out and must log in again. The value for hard timeout must be greater than the value for soft timeout.

-aa Attributes to add, which will be inherited by all users in the group. (When providing multiple attributes, separate `<name>:<value>` pairs for attributes with commas. When providing multiple values for an attribute, separate those values with pipe symbols (`|`). Enclose the group of attributes in quotation marks. For example: `-aa "one:1|2|3,two:2"`.) Be aware that attributes can be defined at the realm, group, and user levels, which means the potential exists for attribute conflicts. See [Appendix B, "Attribute Conflict-Resolution Rules,"](#) for information on how these conflicts are resolved.

-ra Attributes to remove, which will remove these previously inherited attributes from all users in the group. To remove attributes, only the attribute names are required. (When providing multiple attributes, separate them with commas and enclose the group of attributes in quotation marks.)

- ar Role(s) to add, which will be inherited by all users in the group. (When providing multiple roles, separate the roles with commas and enclose the group of roles in quotation marks.)
- rr Role(s) to remove, which will remove these previously inherited roles from all users in the group. (When providing multiple roles, separate the roles with commas and enclose the group of roles in quotation marks.)

Restrictions: If adding roles, the roles must already have been created.

modify_realm

Description: The `modify_realm` command modifies or adds information for a security realm.

Command-Specific Options:

- rlid Unique realm ID (realm name). Mandatory option.
- sdescr Short description of the realm.
- descr Description of the realm.
- aa Attributes to add, which will be inherited by all users in the realm, regardless of group association. (When providing multiple attributes as values, separate `<name>: <value>` pairs for attributes with commas. When providing multiple values for an attribute, separate those values with pipe symbols (`|`). Enclose the group of attributes in quotation marks. For example: `-aa "one:1|2|3,two:2"`.) Be aware that attributes can be defined at the realm, group, and user levels, which means the potential exists for attribute conflicts. See [Appendix B, "Attribute Conflict-Resolution Rules,"](#) for information on how these conflicts are resolved.
- ra Attributes to remove, which will remove these previously inherited attributes from all users in the realm. To remove attributes, only the attribute names are required. (When providing multiple attributes, separate them with commas and enclose the group of attributes in quotation marks.)
- op Operation to perform on the security realm's password policy or portions of the policy. Valid values are `modify` and `remove`. (The `modify` operation changes the portions of the password policy that you specify to the values you provide. The `remove` operation deletes the realm's custom password policy and reverts to the default password policy. [Table 4, "Default Password Policy,"](#) on page 18 describes the default password policy.)
- plen Minimum password length (number of characters). Must be greater than 1.
- mxlen Maximum password length (number of characters). Must be greater than or equal to the minimum length.
- ac Minimum number of alphabetic characters required in a password.
- al Minimum number of the alphabetic characters that must be lowercase.
- au Minimum number of the alphabetic characters that must be uppercase.
- oc Minimum number of other characters (numbers or special characters) required in a password.
- md Minimum number of characters in a new password that must be different from characters in the old password.
- mna Minimum password age, in weeks, before a password can be changed. Mandatory option if `-mxa` is specified.
- mxa Maximum password age, in weeks, at which time the user is notified to change the password. Mandatory option if `-mna` is specified.

- mxex Maximum expiration threshold, which is the number of weeks beyond the maximum age that a password can be changed by the user. After that, it must be reset by the security administrator.
- hiex Number of weeks in which a user cannot reuse a previous password.
- hisz Number of previous passwords that a user cannot reuse.
- mxr Maximum number of consecutive failed login attempts (maximum retries) before a user account is locked.
- lkitr Lock interval, which is the time span (in minutes) during which consecutive failed login attempts are counted in determining whether to lock a user account.
- dl Dictionary list containing comma-separated words that are prohibited as user passwords for the security realm. Enclose the list of words in quotation marks. (During evaluation of a submitted password, the dictionary list is checked first. If an exact match is found in the dictionary list, no further checking of password constraints occurs and the password cannot be used. If a match is not found in the dictionary list, then the password is evaluated against other parts of the password policy.)

Restrictions: None, but be cautious in removing the password policy or changing such values as minimum password length, number of alphabetic characters, and number of other characters. Otherwise, current passwords defined for user accounts in the realm might no longer be valid.

modify_user

Description: The `modify_user` command modifies or adds information for a user account in the specified security realm.

Command-Specific Options:

- uid Unique user ID. Mandatory option.
- rlid Realm ID (realm name) where the user account is defined. Mandatory option.
- fn First name of the user.
- ln Last name of the user.
- ph Phone number of the user.
- xe Phone extension of the user.
- mn Middle name of the user.
- dept Department of the user.
- aa User attributes to add. A user attribute is a custom attribute such as dealer ID or reseller ID. (When providing multiple attributes, separate `<name>: <value>` pairs for attributes with commas. When providing multiple values for an attribute, separate those values with pipe symbols (`|`). Enclose the group of attributes in quotation marks. For example: `-aa "one:1|2|3,two:2"`.) Be aware that attributes can be defined at the realm, group, and user levels, which means the potential exists for attribute conflicts. See [Appendix B, "Attribute Conflict-Resolution Rules,"](#) for information on how these conflicts are resolved.
- ra User attributes to remove. To remove attributes, only the attribute names are required. (When providing multiple attributes, separate them with commas and enclose the group of attributes in quotation marks.)
- ag Groups to add for the user. (When providing multiple groups, separate the groups with commas and enclose the set of groups in quotation marks.)
- rg Groups to remove for the user. (When providing multiple groups, separate the groups with commas and enclose the set of groups in quotation marks.)
- lock Lock status for the user account. Valid values are `true` and `false`.

-pgroup Priority group to add or modify, which must be one of the groups specified for the user. The priority group serves to resolve attribute conflicts and to establish the soft timeout (session inactivity timeout) and hard timeout (maximum session duration timeout) values for user login sessions.

-acctstate User account state. Valid values are `ENABLED` and `DISABLED`. (Enabled means the account will be purged after a certain number of days of inactivity. Disabled means the account will not be purged because of inactivity. See [“Viewing Purged Inactive User Accounts” on page 28](#) for more details.)

-email Email address of the user.

Restrictions: The user password cannot be changed with this command. Instead, the password must be reset. See [“reset password” on page 129](#). (Changing your own password, as a logged-in Management Shell user, requires a different method. See [“change password” on page 123](#).)

remove_group

Description: The `remove_group` command deletes a realm group for the specified security realm. Any group attributes and roles, which are inherited by users in the group, will no longer be associated with those users once the group is deleted. If the group is the priority group for any users, reassign the users to a different priority group before deleting the group. If you do not reassign users to a different priority group, the `DEFAULT` group for the security realm becomes their priority group. (You can reassign users after deleting the group, but it is less disruptive to do this beforehand.)

Command-Specific Options:

-rlid Unique realm ID (realm name) where the group is defined. Mandatory option.

-gid Unique group ID of the group to be deleted. Mandatory option.

Restrictions: None.

remove_realm

Description: The `remove_realm` command deletes a security realm, and all users and groups defined for that realm.

Command-Specific Options:

-rlid Unique realm ID (realm name) of the realm to be deleted. Mandatory option.

Restrictions: The `UPSEC` security realm cannot be deleted. That realm contains the administrator user and users who can perform actions on the Security Server and Unified Platform.

remove_role

Description: The `remove_role` command deletes a security role and removes the role's association with any groups. This removes access privileges provided by the role (via its use in authorization policies) for users in the affected groups.

Command-Specific Options:

-roid Unique role ID (role name) of the role to be deleted. Mandatory option.

Restrictions: The `ADMIN` role cannot be deleted. That role is for the security administrator and other users with administrative privileges.

remove_user

Description: The `remove_user` command deletes a user account from the specified security realm and removes that user's association from security realm groups.

Command-Specific Options:

-rlid Unique realm ID (realm name) where the user account is defined. Mandatory option.

-uid Unique user ID for the user account. Mandatory option.

Restrictions: The `secadmin` user account cannot be deleted. That is the account for the security administrator.

reset_password

Description: The `reset_password` command resets the password for a user account to a randomly generated password, which is displayed. This password complies with the password policy for the security realm where the user account is defined. After the password has been reset, the user is required to change this password at the next login.

Command-Specific Options:

-rlid Unique realm ID (realm name) where the user account is defined. Mandatory option.

-uid User ID for the user account. Mandatory option.

Restrictions: None.

unlock_user

Description: The `unlock_user` command unlocks a locked user account, enabling logins.

Command-Specific Options:

-uid Unique user ID. Mandatory option.

-rlid Unique realm ID (realm name) for the user account. Mandatory option.

Restrictions: None.

Policy Management Commands

The following are policy management commands, arranged in alphabetical order.

create_auth_policy

Description: The `create_auth_policy` command creates a new authorization policy associated with the specified security realm. After an authorization policy is created, it must be published to activate it for use by the target application.

Command-Specific Options:

-id Unique policy ID (policy name). Mandatory option. The first few characters of the policy name must be a realm name that identifies the realm to which the policy applies, such as `PC_<SomeName>` for a Product Catalog policy.

-description Description of the policy.

-realm Realm ID (realm name) for the policy. Mandatory option.

-combid Rule-combining algorithm for the policy used to resolve rule conflicts. Valid values are `permit-overrides`, `deny-overrides`, and `first-applicable` (`permit-overrides` means that if any rule evaluates to `Permit`, the final authorization decision is also `Permit`; `deny-overrides` means that if any rule evaluates to `Deny`, the

final authorization decision is also `Deny`; first-applicable means that the result of the first relevant rule encountered is the final authorization decision). If this option is not provided, the default is `permit-overrides`.

`-rules` List of rules for the policy. Mandatory option. (When providing multiple rules as values, separate the rules with commas and enclose the group of rules in quotation marks.)

`-b` Batch policy loading. When this option is specified, no other options are recognized. By default, batch policies are loaded from the `$JBOSS_HOME/batch/policy` directory. An alternate directory can be used. For details, see [“Bulk Policy Operations” on page 45](#).

Restrictions: Unless the batch load option is used, the security realm and the authorization rules must already have been created.

create_auth_rule

Description: The `create_auth_rule` command creates a new authorization rule for use in authorization policies. For creation of many rules, the recommended approach is by means of bulk policy loading by a batch process. See [“Bulk Policy Operations” on page 45](#) for information.

Command-Specific Options:

`-id` Unique rule ID (rule name). Mandatory option.

`-description` Description of the rule.

`-subject` Subject(s) of the rule. Although subjects can be many things, currently Comverse ONE applications use roles as the subjects of rules. (When providing multiple subjects as values, separate the subjects with commas and enclose the group of subjects in quotation marks.) If this option is not provided, the default is `Any`.

`-resource` Resource(s) of the rule. Resources are application-specific and might be data, service, or system components. For example, service methods can be resources. (When providing multiple resources as values, separate the resources with commas and enclose the group of resources in quotation marks.) If this option is not provided, the default is `Any`.

`-action` Action(s) of the rule. An action is an operation on a resource. The actions are application-specific. Some examples are `invoke`, `enable`, `disable`, `read`, `write`, `create`, `update`, `delete`, and so on. (When providing multiple actions as values, separate the actions with commas and enclose the group of actions in quotation marks.) If this option is not provided, the default is `Any`.

`-effect` The intended consequence of the satisfied rule. Valid values are `Permit` and `Deny`. If this option is not provided, the default is `Permit`.

Restrictions: None.

list_auth_policy

Description: The `list_auth_policy` command shows the details of all authorization policies, or an individual policy.

Command-Specific Options:

`-id` Unique policy ID (policy name) of an individual authorization policy or the common prefix for multiple policies. (For example, if multiple policies are named `ABC<SOMETHING>`, `ABC` is the prefix.)

Restrictions: None.

list_auth_rule

Description: The `list_auth_rule` command shows the details of authorization rules. If the `-pid` option is provided, the command shows the list of rules for the specified policy or policies instead of the normal authorization rule details.

Command-Specific Options:

- `-id` Unique rule ID (rule name) of an individual authorization rule, or the common prefix for multiple rules. (For example, if multiple rules are named `ABC<SOMETHING>`, `ABC` is the prefix.)
- `-pid` Unique policy ID (policy name) of an individual authorization policy, or the common prefix for multiple policies. (For example, if multiple policies are named `ABC<SOMETHING>`, `ABC` is the prefix.)

Restrictions: None.

modify_auth_policy

Description: The `modify_auth_policy` command adds or modifies information for an authorization policy in the Security Server database. If the policy has already been published, and changes are made to anything except the policy description, the policy must be republished to activate it for use by the target application.

Command-Specific Options:

- `-id` Unique policy ID (policy name). Mandatory option.
- `-description` Description of the policy.
- `-realm` Realm ID (realm name) for the policy.
- `-combid` Rule-combining algorithm for the policy used to resolve rule conflicts. Valid values are `permit-overrides`, `deny-overrides`, and `first-applicable` (`permit-overrides` means that if any rule evaluates to `Permit`, the final authorization decision is also `Permit`; `deny-overrides` means that if any rule evaluates to `Deny`, the final authorization decision is also `Deny`; `first-applicable` means that the result of the first relevant rule encountered is the final authorization decision).
- `-rules` List of rules for the policy. (When providing multiple rules as values, separate the rules with commas and enclose the group of rules in quotation marks.)

Restrictions: None.

modify_auth_rule

Description: The `modify_auth_rule` command adds or modifies information for an authorization rule in the Security Server database. If the rule is used in a policy that has already been published, and changes are made to anything except the rule description, the policy must be republished to activate it for use by the target application.

Command-Specific Options:

- `-id` Unique rule ID (rule name). Mandatory option.
- `-description` Description of the rule.
- `-subject` Subject(s) of the rule. Although subjects can be many things, currently Comverse ONE applications use roles as the subjects of rules. (When providing multiple subjects as values, separate the subjects with commas and enclose the group of subjects in quotation marks.)
- `-resource` Resource(s) of the rule. Resources are application-specific and might be data, service, or system components. For example, service methods can be resources. (When providing multiple resources as values, separate the resources with commas and enclose the group of resources in quotation marks.)

-action Action(s) of the rule. An action is an operation on a resource. The types of actions are application-specific. Some examples are invoke, enable, disable, read, write, create, update, delete, and so on. (When providing multiple actions as values, separate the actions with commas and enclose the group of actions in quotation marks.)

-effect The intended consequence of the satisfied rule. Valid values are Permit and Deny.

Restrictions: None.

publish_policy

Description: The `publish_policy` command publishes an authorization policy, or multiple policies, from the Security Server database to XACML format. That is, it exports policies from the database to XML files that conform to the XACML schema and places the files in the `$JBoss_HOME/conf/policy` directory on the Security Server. If a manager option is provided, the command also resynchronizes policies on the applicable nodes. This means that the Unified Platform Agent (UPA) residing on the same physical node as the application to which the policy applies retrieves the policy on demand (without requiring a UPA restart) so that the UPA can load the newly published XACML-formatted policy. Once the policy is loaded, authorization requests use the new policy.

Activities such as publishing policies and resynchronizing policies rely on a policy-naming convention. The first few characters of the policy name must be a realm name that identifies the realm to which the policy applies, such as `PC_<SomeName>` for a Product Catalog policy.

Command-Specific Options:

-id Unique policy ID (policy name), or the common prefix for multiple policies. (For example, if multiple policies are named `ABC<SOMETHING>`, `ABC` is the prefix.)

Manager Options:

-c Class (node type) of policies, such as SDP or SAPI. Resynchronizes all policies that match the -id and belong to the specified class.

-mon Managed object name (that is, a node name). Resynchronizes all policies that match the -id for the specified node.

-n Node instance (that is, the IP address of the instance). Resynchronizes all policies that match the -id for the specified node instance.

Restrictions: None.

remove_auth_policy

Description: The `remove_auth_policy` command deletes authorization policies from the Security Server database. If a deleted policy has already been published, then policies must be published again, using the `publish_policy` command with the appropriate manager option, to resynchronize policies for the target node(s).

Command-Specific Options:

-id Unique policy ID (policy name) of an individual authorization policy, or the prefix for a group of policies. (For example, if multiple policies are named `ABC<SOMETHING>`, `ABC` is the prefix.) Mandatory option.

Restrictions: None.

remove_auth_rule

Description: The `remove_auth_rule` command deletes authorization rules from the Security Server database and from all authorization policies in the database. If a policy that

uses the rule has already been published, the policy must be republished so that the change takes effect for the target application.

Command-Specific Options:

-id Unique rule ID (rule name) of an individual authorization rule, or the common prefix for multiple rules. (For example, if multiple rules are named *ABC<SOMETHING>*, *ABC* is the prefix.) Mandatory option.

Restrictions: None.

resync_policy

Description: The `resync_policy` command resynchronizes all authorization policies of the Uniform Platform Agents (UPAs) with the appropriate policies from the Security Server, or policies for specified nodes. This means that, on demand and without a UPA restart, UPAs retrieve the XACML-formatted authorization policies (that is, published policies or manually created policies) from the Security Server for the applicable node(s). The UPA(s) then load the updated policies for use by target applications.

Command-Specific Options:

None.

Manager Options:

-c Class (node type) of policies, such as SDP or SAPI. Resynchronizes policies that belong to the specified class.

-mon Managed object name (that is, a node name). Resynchronizes policies for the specified node.

-n Node instance (that is, the IP address of the instance). Resynchronizes policies for the specified node instance.

Restrictions: None.

Audit Management Commands

There are currently no commands used exclusively for audit management. However, the `build_report` command (with `audit` as the report type) allows you to query audit records and view the results in a report. The `build_report` command is used for many other purposes (see the *Unified Platform Guide*), but the following information about the command is specific to reports for audit records.

build_report

Description: The `build_report` command generates a report of the specified type (audit). If no beginning and ending dates are provided, the report includes audit records for the last hour.

Command-Specific Options:

-r Report type to build. Mandatory option. For reports on audit records, the value must be `audit`.

-b Beginning date, in "MM/DD/YYYY" format, for the reporting time period.

-e Ending date, in "MM/DD/YYYY" format, for the reporting time period.

-uid Unique user ID, which limits the report to audit records for only a specific user.

-cn Command name, which limits the report to audit records corresponding to a particular command (such as `login`).

-xid External ID, which limits the report to audit records corresponding to a particular external ID (such as account number or subscriber number).

Restrictions: If beginning dates and ending dates are provided, there cannot be more than one day's difference between those dates.

Key Management Commands

The following are key management commands (for symmetric encryption keys), arranged in alphabetical order.

create_key

Description: The `create_key` command creates a symmetric encryption key in the Security Server database.

Command-Specific Options:

-kid Unique key ID. The key ID can be a well-known custom name that one or more components use to look up the key or a unique global key ID (GKID) generated by the Security Server. A custom name can contain letters, digits, and special characters (such as underscore and hyphen). If this option is not provided, the Security Server creates the key ID. If the Security Server creates the ID, the format is <SSID>-<KID>, where <SSID> is the Security Server ID and <KID> is a series of digits, with the combination resulting in a unique global key ID. Examples are 01-16343162, 02-18570884, and so on.

-algo Algorithm for which the key is being created. Valid values are AES and Blowfish. If this option is not provided, the default is AES.

Restrictions: None.

delete_key

Description: The `delete_key` command deletes a symmetric encryption key in the Security Server database.

Command-Specific Options:

-kid Key ID. Mandatory option.

Restrictions: The status of the key must be inactive before it can be deleted. See [“disable_key”](#) below.

disable_key

Description: The `disable_key` command changes the status of a symmetric encryption key in the Security Server database to inactive (that is, not usable).

Command-Specific Options:

-kid Key ID. Mandatory option.

Restrictions: None.

enable_key

Description: The `enable_key` command changes the status of a symmetric encryption key in the Security Server database from inactive to active.

Command-Specific Options:

-kid Key ID. Mandatory option.

Restrictions: None.

list_keys

Description: The `list_keys` command lists all the symmetric encryption keys in the Security Server database.

Command-Specific Options: None

Restrictions: None.

Credentials Management Commands

The following are credentials management commands, arranged in alphabetical order.



NOTE

Currently, credentials are limited to database passwords for business databases used by Comverse ONE applications and SNMP community strings for network devices.

list_credential

Description: The `list_credential` command lists credentials (database passwords or SNMP community strings for network devices) stored in the Security Server database.

Command-Specific Options:

`-type` Credential type. Currently, `database` and `network` are the only valid values. If this option is not provided, the default is `database`.

Restrictions: None.

publish_credential

Description: The `publish_credential` command propagates the credential (database password created with the `store_credential` command) to the target database instance on the local node. If a manager option is provided, the command propagates the password to the target database instance on remote node(s). When the `-rc` option is used, the password cache on the node(s) is updated. Publishing credentials is a two-step process and the `publish_credential` command must be executed twice, first without the `-rc` option and then with the `-rc` option.

Command-Specific Options:

`-type` Credential type. Mandatory option. Currently, `database` is the only valid value for this option.

`-uid` Unique database user ID. Mandatory option.

`-dbtype` Database type. Examples of some valid values are `RATING`, `PCAT`, `CBS`, and so on. If this option is not provided, the default is `CBS`.

`-in` Database instance name. Mandatory option. Examples of valid values are `MAIN` or `HIST` (if the database type is `RATING`).

`-rc` Refresh cache. This option is valid only if the credential type is `database`. If this option is used, the password cache on the node(s) is updated.

Manager Options:

- c Class (node type) where the database resides, such as SDP. Propagates the password for the specified username to the target database instance on nodes of the given type. When the -rc option is used, updates the password cache on the nodes.
- mon Managed object name (that is, a node name). Publishes the password for the specified username to the target database instance on the given named node. When the -rc option is used, updates the password cache on the node.
- n Node instance (that is, the IP address of the instance). Publishes the password for the specified username to the target database instance on the given node instance. When the -rc option is used, updates the password cache on the node.

Restrictions: Publishing credentials should be done during a maintenance window.

remove_credential

Description: The `remove_credential` command deletes a credential (database password or SNMP community string for a network device) previously stored in the Security Server database. This command is used, in conjunction with the `store_credential` command, to change existing database or network credentials. If a database password has already been published (that is, propagated to the target database), it remains in effect for the target database until credentials are published again.

Command-Specific Options:

- type Credential type. Mandatory option. Currently, `database` and `network` are the only valid values.
- uid Unique database user ID. Mandatory option if the credential type is `database`.
- dbtype Database type. Valid only if the credential type is `database`. Examples of some valid values are `RATING`, `PCAT`, `CBS`, and so on. If this option is not provided, the default is `CBS`.
- in Database instance name. Mandatory option if the credential type is `database`. Examples of valid values are `MAIN` or `HIST` (if the database type is `RATING`).
- nc Node class of the network device. Mandatory option if the credential type is `network`.
- nm Node name of the network device. Mandatory option if the credential type is `network`.
- ni Node instance of the network device. Mandatory option if the credential type is `network`.

Restrictions: None.

store_credential

Description: The `store_credential` command stores a credential (database password or SNMP community string for a network device) in the Security Server database.

Command-Specific Options:

- type Credential type. Mandatory option. Currently, `database` and `network` are the only valid values.
- uid Unique database user ID. Mandatory option if the credential type is `database`.
- dbtype Database type. Valid only if the credential type is `database`. Examples of some valid values are `RATING`, `PCAT`, `CBS`, and so on. If this option is not provided, the default is `CBS`.
- in Database instance name. Mandatory option if the credential type is `database`. Examples of valid values are `MAIN` or `HIST` (if the database type is `RATING`).

- pwd Password for the database user ID to be stored. Mandatory option if the credential type is database. A database user (identified by a user ID, such as CBS_OWNER) must have the same password for a given instance across all databases of a given type in the system.
- nc Node class of the network device. Mandatory option if the credential type is network.
- nm Node name of the network device. Mandatory option if the credential type is network.
- ni Node instance of the network device. Mandatory option if the credential type is network.
- comm SNMP community string for the network device.
- b Batch loading for database credentials (database passwords). When this option is specified, no other options are recognized. By default, source files that contain database passwords are loaded from the \$JBOSS_HOME/batch/credentials directory. An alternate directory can be used. For details, see [“Bulk Credentials Operations” on page 70](#).

Restrictions: A database user (identified by a user ID, such as CBS_OWNER) must have the same password for a given instance across all databases of a given type in the system.

Chapter 8

Database Reference

8

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

Overview

This chapter contains a detailed description of the Security Server database tables (that is, the XE database instance). These table descriptions are in alphabetical order and include the following information:

- Table name and a brief description of table
- List of tables that depend on the table
- Descriptions of triggers, check constraints, foreign keys, and indexes defined on the table
- Descriptions of every field in the table

Each field description includes the following information:

- **Field name**
- **Data type and size**
- **Comments:** Brief field description
- **Values:** Rules that constrain field values (preceded by R:) and default values for the field (preceded by D:)
- **Null:** Indicates whether a Null value is allowed in the field



NOTE

When you insert data, including a single quote (') in a text string can cause an error.



NOTE

A range is defined by its endpoints. Date ranges in the Comverse ONE solution consist of two endpoints: an active date and an inactive date. Any date is within a particular defined range if the date is on or past the range's active date and before (but not including) the range's inactive date. The range includes the starting point, but excludes the ending point.

Every field description in this document contains either one of the documented data types listed below or an Oracle data type. [Table 14](#) lists the valid documented data types and their corresponding relational database data types for Oracle and Sybase.

Table 14 Database Field Types

Documented Type	Oracle Type	Sybase Type	Minimum Value	Maximum Value
bit	Number(1,0)	bit (1 byte)	0	1
char[N]	char(N)	char[N]	N characters	N characters
Fixed-length string, padded with blanks if smaller than N. N cannot exceed 255.				
datetime	date	datetime (8 bytes)	varies	varies
For Oracle, stores date/time between January 1, 4712 B.C., and December 31, 4712. For Sybase, stores date/time between January 1, 1753, and December 31, 9999.				
image	blob	image (16 bytes)	—	—
For Oracle, stores variable-length binary data up to 2 GB. For Sybase, stores a pointer to a memory space allocated in multiples of 2 KB that cannot exceed 2 GB.				
int	Number(10,0)	int (4 bytes)	-2,147,483,648 (-2 ³¹)	2,147,483,647 (2 ³¹ - 1)
numeric	Number(18,0)	numeric (10) (8 bytes)	-999,999,999, 999,999,999	999,999,999, 999,999,999
smalldt	date	smalldatetime (4 bytes)	varies	varies
For Oracle, stores date/time between January 1, 4712 B.C., and December 31, 4712. For Sybase, stores date/time between January 1, 1900, and June 6, 2079.				
smallint	Number(6,0)	smallint (2 bytes)	-32,768 (-2 ¹⁵)	32,767 (2 ¹⁵ - 1)
text	varchar2(4000)	text (16 bytes)	—	—
For Oracle, stores variable-length character data up to 4,000 bytes. For Sybase, stores a pointer to a memory space allocated in multiples of 2 KB that cannot exceed 2 GB.				
tinyint	Number(3,0)	tinyint (1 byte)	0	255
varchar[N]	varchar2(N)	varchar[N]	0 characters	N characters
Variable-length string. N cannot exceed 255.				

[Table 14](#) also includes minimum and maximum values for each documented type. These are not necessarily the minimum and maximum supported by your database; rather, they are the minimum and maximum values supported by the Comverse ONE solution for fields of this type. Therefore, you should not enter field values that exceed the minimum or maximum values documented here, even if the corresponding RDB data type allows those values.

The data type column for char and varchar fields can display two values:

- Character limit
- Physical size

The value on the right is the character limit of the field. The character limit of the field indicates the maximum number of characters the field can hold.



Do not enter values with sizes larger than the character limit listed. Doing so might result in corrupt data.

The value in parentheses is the physical size of the field. The physical size of the field is the number of bytes needed to store a field.

In the example below, the character limit for short_display is 5 characters and the physical size of the field is 15 bytes.

Field Name	Data Type		Comments	Values	Null
short_display	varchar (15)	5	Short text description for ABI source.		Null

SEC_AA_EVENT

Security auditing and accounting table. Contains all the audit records for a given client organization site.

Indexes: primary-key on aud_id

Field Name	Data Type		Comments	Values	Null
aud_id	number	18	ID of this audit record.		
hdr_rec_len	varchar2	6	Length of the record (hexadecimal value), from the audit record header.		
hdr_time_offset	date		Time offset, from the audit record header. Represents the number of seconds since the beginning of the epoch, as defined by the Single UNIX Standard (midnight on January 1, 1970).		
hdr_event_outcome	number	10	Event outcome code, from the audit record header. (The value is a decimal representation of the hexadecimal event outcome code.) The code comes from the set of standardized event outcome codes as defined by the Distributed Audit Service (XDAS) specification.		
hdr_event_number	number	10	Event number, from the audit record header. (The value is a decimal representation of the hexadecimal event number.) The number is part of a generic event taxonomy, a set of semantic meanings for events that are likely to meet most applications' auditing needs.		
orig_loc_name	varchar2	512	Name of the host or service that is reporting the auditable event.		Null
orig_loc_addr	varchar2	512	Communication service end-point address.		Null
orig_service_type	varchar2	512	Protocol used by the originator location address.		Null
orig_authen_auth	varchar2	512	Name of a server, or domain and realm, that provides the identities involved in the associated events.		Null
orig_principal_name	varchar2	512	User name relative to the originator authentication authority.		Null
orig_principal_id	varchar2	512	User ID of the principal, relative to the authentication authority.		Null
init_authen_auth	varchar2	512	Host, service, or domain and realm name of the authentication service that provides the identity attributes for the initiator of an auditable action.		Null
init_domain_name	varchar2	512	Name of the user or ID entity that is initiating an auditable action, relative to the initiator authentication authority.		Null
init_domain_id	varchar2	512	Identifier (user ID, globally unique ID, and so forth) of the user or identity that is initiating the auditable action, relative to the initiator authentication authority.		Null

Field Name	Data Type		Comments	Values	Null
tgt_loc_name	varchar2	512	Name of the target resource, host, or service of the event that is being reported.		Null
tgt_loc_addr	varchar2	512	Communication service end-point address, which can be a URL or other type of address that fully specifies a connection point.		Null
tgt_service_type	varchar2	512	Protocol used by the target location address.		Null
tgt_authen_auth	varchar2	512	Name of a server, or domain and realm, that provides the identities involved in the associated events.		Null
tgt_principal_name	varchar2	512	User name relative to the target authentication authority.		Null
tgt_principal_id	varchar2	512	User ID of the target principal, relative to the authentication authority.		Null
aud_event_source	varchar2	512	Not currently used.		Null
aud_event_info	varchar2	512	Carries additional information that is specific to the application domain.		
external_id	varchar2	512	Unique ID (for example, a subscriber ID) that can be used to track the activity through the system.		Null

SEC_DPM_PASSWORD

Security database password management table. Contains the database passwords for the various databases of the client applications.

Indexes: primary-key on db_id

Field Name	Data Type		Comments	Values	Null
db_id	number	18	ID of this record		
user_id	varchar2	40	User ID for the database.		
db_type	varchar2	40	Type of database, such as RATING, PCAT, CBS, and so on.		
instance	varchar2	40	Name of the instance of the database (for example, MAIN or HIST if the database type is RATING).		
password	varchar2	40	Password for the database instance. The password is encrypted before being stored in the database.		

SEC_IDM_COUNTER

Security identity management table. Contains the counter value for each login attempt by a user, which is used for the user account lock feature.

Indexes: primary-key on iu_cid
 unique on iu_user

Foreign Keys: (iu_user) must exist in SEC_IDM_USER(iu_id)

Field Name	Data Type		Comments	Values	Null
iu_cid	number	18	ID of this record.		
iu_count	number	5	Counter that reflects the number of the user's login attempts.		
iu_last_access_time	number	38	Last access time for the user.		
iu_user	number	18	ID of the user, from SEC_IDM_USER.iu_id.		

SEC_IDM_GROUP

Security identity management table. Contains definitions of groups within security realms.

Dependents: SEC_IDM_GROUP_ATTRIBUTE, SEC_IDM_GROUP_ROLE, SEC_IDM_USER, SEC_IDM_USER_GROUP

Indexes: primary-key on ig_id
unique on (grp_name, ig_realm_id, ig_default)

Foreign Keys: (ig_realm_id) must exist in SEC_IDM_REALM(irl_id)
(ig_policy_id) must exist in SEC_SESS_POLICY(sp_id)

Field Name	Data Type		Comments	Values	Null
ig_id	number	18	ID of this record.		
grp_name	varchar2	200	Unique group name.		
language_code	number	11	Language code, from LANGUAGE_CODE_REF.		Null
ig_short_display	varchar2	30	Short text description for this record.		Null
ig_display_value	varchar2	240	Text description to be displayed on screen.		Null
ig_update_count	number	18	Number of times this record has been updated.		Null
ig_create_dt	date		Date and time this record was created.		
ig_modify_dt	date		Date and time this record was last modified.		
ig_modify_who	varchar2	30	ID of the user who last changed this record.		Null
ig_realm_id	number	18	ID of the realm that this group is associated with, from SEC_IDM_REALM.irl_id.		Null
ig_policy_id	number	18	ID of the session policy for this group, from SEC_SESS_POLICY.sp_id.		Null
ig_default	varchar2	1	Indicates whether this is the default group (Y/N).	D: 'Y'	Null

SEC_IDM_GROUP_ATTRIBUTE

Security identity management table. Contains attributes and corresponding values for groups.

Indexes: This table has no indexes.

Foreign Keys: (iga_grp_id) must exist in SEC_IDM_GROUP (ig_id)

Field Name	Data Type		Comments	Values	Null
iga_grp_id	number	18	ID of the group, from SEC_IDM_GROUP.ig_id.		
iga_attribute_name	varchar2	50	Name of the group attribute.		Null
iga_attribute_val	varchar2	200	Value of the group attribute.		Null

SEC_IDM_GROUP_ROLE

Security identity management table. Contains roles for groups.

Indexes: unique on (iug_grp_id,iug_role_id)

Foreign Keys: (iug_grp_id) must exist in SEC_IDM_GROUP(ig_id)
(iug_role_id) must exist in SEC_IDM_ROLE(ir_id)

Field Name	Data Type		Comments	Values	Null
iug_grp_id	number	18	ID of the group, from SEC_IDM_GROUP.ig_id.		Null
iug_role_id	number	18	ID of the role, from SEC_IDM_ROLE.ir_id.		Null

SEC_IDM_PURGED_USERS

Security identity management table. Contains the users removed because of inactivity. All records in the table are deleted when the number of records equals the configured maximum number (the default is 50).

Indexes: unique on (iu_id)

Field Name	Data Type		Comments	Values	Null
iu_id	number	18	ID of this record.		
user_id	varchar2	200	User ID for the user.		
realm_name	varchar2	200	Realm name where the user account was defined.		
last_login_date	date		Last date the user logged in.		Null
iu_acct_state	number	1	State of the user account. For all records in this table, the account state is 0 (that is, enabled).		Null

SEC_IDM_REALM

Security identity management table. Contains security realms for users. A realm is also known as a login domain.

Dependents: SEC_IDM_GROUP, SEC_IDM_REALM_ATTRIBUTE, SEC_IDM_USER

Indexes: primary-key on irl_id
unique on realm_name

Foreign Keys: (irl_policy_id) must exist in SEC_PASS_POLICY(policy_id)

Field Name	Data Type		Comments	Values	Null
irl_id	number	18	ID of this record.		
realm_name	varchar2	200	System name for this security realm. Security realm typically represents a type of login domain.		
language_code	number	11	Language code, from LANGUAGE_CODE_REF.		Null
irl_short_display	varchar2	30	Short text description for this record.		Null
irl_display_value	varchar2	240	Text description to be displayed on screen.		Null
irl_update_count	number	18	Number of times this record has been updated.		Null
irl_create_dt	date		Date and time this record was created.		
irl_chg_dt	date		Date and time this record was last modified.		
irl_chg_who	varchar2	30	ID of user who last changed this record.		Null
irl_policy_id	number	18	ID of the password policy for this security realm, from SEC_PASS_POLICY.policy_id.		Null

SEC_IDM_REALM_ATTRIBUTE

Security identity management table. Contains attributes for security realms.

Indexes: This table has no indexes.

Foreign Keys: (irl_id) must exist in SEC_IDM_REALM(irl_id)

Field Name	Data Type		Comments	Values	Null
irl_id	number	18	ID of the security realm, from SEC_IDM_REALM.irl_id.		Null
irl_attribute_name	varchar2	50	Name of the security realm attribute.		Null
irl_attribute_val	varchar2	200	Value of the security realm attribute		Null

SEC_IDM_ROLE

Security identity management table. Contains the roles that groups of users can perform within the confines of an application. Role can be defined as an abstract entity with privileges to access resources.

Dependents: SEC_IDM_GROUP_ROLE

Indexes: primary-key on ir_id
unique on role_name

Field Name	Data Type		Comments	Values	Null
ir_id	number	18	ID of this record.		
role_name	varchar2	200	Unique role name.		
language_code	number	11	Language code, from LANGUAGE_CODE_REF.		Null
ir_short_display	varchar2	30	Short text description for this record.		Null
ir_display_value	varchar2	240	Text description to be displayed on screen.		Null
ir_update_count	number	18	Number of times this record has been updated.		Null
ir_create_dt	date		Date and time this record was created.		
ir_chg_dt	date		Date and time this record was last modified.		
ir_chg_who	varchar2	30	ID of the user who last changed this record.		Null

SEC_IDM_USER

Security identity management table. Contains all users of the system who need to access the resources defined within the security system.

Dependents: SEC_IDM_COUNTER, SEC_IDM_USER_ATTRIBUTE, SEC_IDM_USER_GROUP, SEC_IDM_USER_SESSION, SEC_PASS_POLICY_HIST

Indexes: primary-key on iu_id
unique on (user_id, iu_rlid)

Foreign Keys: (iu_rlid) must exist in SEC_IDM_REALM(irl_id)

Field Name	Data Type		Comments	Values	Null
iu_id	number	18	ID of this record.		
user_id	varchar2	200	User ID of the application user. Application user must use this identification when logging in to the system.		
iu_rlid	number	18	ID of the user's security realm, from SEC_IDM_REALM.irl_id.		Null
iu_first_name	varchar2	200	User's first name.		
iu_middle_name	varchar2	200	User's middle name.		Null
iu_last_name	varchar2	200	User's last name.		
iu_last_login	date		Date and time when the user last logged in to the system.		Null
iu_password	varchar2	500	User's password. The password is encrypted before being stored in the database.		
iu_phone	varchar2	20	User's phone number.		Null
iu_extension	varchar2	10	User's telephone extension.		Null
iu_department	varchar2	30	User's department.		Null
iu_update_count	number	38	Number of times this record has been updated.		Null
iu_create_dt	date		Date and time this record was created.		
iu_chg_dt	date		Date and time this record was last modified.		
iu_chg_who	varchar2	30	ID of user who last changed this record.		Null
iu_lock	varchar2	1	Indicates whether the user account is locked (Y/N).	D: 'N'	Null
iu_email_address	varchar2	50	User's email address.		Null
iu_force_chg_pass	varchar2	1	Indicates whether the user will be forced to change his/her password at the next login (Y/N).	D: 'N'	

Field Name	Data Type		Comments	Values	Null
iu_state	number	1	Used internally to track the user's login state (first login, login after a password reset, and so on).		
iu_priority_grp	varchar2	200	User's priority group, which serves to resolve attribute conflicts and specifies the soft timeout/hard timeout settings for the user's login sessions.		Null
iu_acct_state	number	1	State of the user account, which determines whether it will be purged after a certain number of days of inactivity. 0 = enabled (will be purged) 1 = disabled (will not be purged)		Null

SEC_IDM_USER_ATTRIBUTE

Security identity management table. Contains user attributes that are application-specific. This table is used by applications to store custom attributes (such as dealer ID, reseller ID, and so on).

Indexes: This table has no indexes.

Foreign Keys: (iua_user_id) must exist in SEC_IDM_USER(iu_id)

Field Name	Data Type		Comments	Values	Null
iua_user_id	number	18	ID of the user, from SEC_IDM_USER.iu_id.		
iua_attribute_name	varchar2	50	Name of the user attribute.		Null
iua_attribute_val	varchar2	200	Value of the user attribute.		Null

SEC_IDM_USER_GROUP

Security identity management table. Holds the association of users with groups.

Indexes: unique on (iug_usr_id,iug_grp_id)

Foreign Keys: (iug_grp_id) must exist in SEC_IDM_GROUP(ig_id)
(iug_usr_id) must exist in SEC_IDM_USER(iu_id)

Field Name	Data Type		Comments	Values	Null
iug_usr_id	number	18	ID of the user, from SEC_IDM_USER.iu_id.		Null
iug_grp_id	number	18	ID of the group, SEC_IDM_GROUP.ig_id.		Null

SEC_IDM_USER_SESSION

Security identity management table. Contains all the sessions of all authenticated active users of the system who need access to resources defined within the security system. This table is used to rebuild the active sessions in case the Security Server is restarted or a switchover occurs.

Indexes: primary-key on ius_id
 unique on ius_user_id

Foreign Keys: (ius_user_id) must exist in SEC_IDM_USER(iu_id)

Field Name	Data Type		Comments	Values	Null
ius_id	number	18	ID of this record.		
ius_user_id	number	18	ID of the user for whom the session is being created, from SEC_IDM_USER.iu_id.		
ius_sess_id	varchar2	50	Session ID that is the unique identifier for this session.		
ius_sess_data	clob		Session data for the user.		Null
ius_sess_create_dt	date		Date and time this session was created.		Null
ius_sess_expiry_dt	date		Date and time this session will expire.		Null
ius_sess_max_idle_time	date		Date and time this session will reach the maximum session idle time.		Null
ius_sess_last_acc_time	date		Date and time this session was last accessed.		Null
ius_soft_timer	number	15	Amount of time, in milliseconds, that the session can remain inactive (idle) before the user is automatically logged out and the session is terminated.		Null
ius_hard_timer	number	15	Maximum duration of the session, in milliseconds, after which the user is automatically logged out and the session is terminated.		Null

SEC_KM_KEYS

Security key management table. Contains symmetric encryption keys.

Indexes: primary-key on km_db_id

Field Name	Data Type		Comments	Values	Null
km_db_id	number	18	ID of this record.		
km_id	varchar2	512	Key ID (global key ID or custom name) of the key that needs to be stored.		
key_val	varchar2	2048	Value of the key.		
old_key_val	varchar	2048	Old value of the key, before the key was changed.		Null
key_algo	varchar2	40	Algorithm for which the key was generated.		
key_status	char	3	Status of the key (active, inactive).	D: 'ACT'	
key_length	number	20	Length of the key.		
km_key_create_dt	date		Date and time this record was created.		
km_key_chg_dt	date		Date and time this record was last modified.		

SEC_KM_SERVER_KEYS

Security key management table. Contains the asymmetric keys (public/private keys) for each Security Server instance. The table is populated whenever the Security Server instance is started for the first time.

Indexes: primary-key on kms_db_id
 unique on kms_ss_id

Field Name	Data Type		Comments	Values	Null
kms_db_id	number	18	ID of this record.		
kms_ss_id	varchar2	512	ID of the Security Server. (Each Security Server has a minimum of one entry in this table.)		
public_key	varchar2	2000	Public key of the Security Server.		
private_key	varchar2	2000	Private key of the Security Server.		
kms_key_create_dt	date		Date and time this record was created.		

SEC_NETWORK_DEV_CRED

Security network-device credentials table. Contains SNMP community strings for the various network devices. A community string is like a password for a network device and is used for authentication purposes.

Indexes: primary-key on db_id

Field Name	Data Type		Comments	Values	Null
id	number	18	ID of this record.		
node_name	varchar2	50	Node name of the network device.		Null
node_class	varchar2	50	Node class of the network device.		Null
node_inst	varchar2	50	Node instance of the network device.		Null
community	varchar2	50	SNMP community string for the network device. The string is encrypted before being stored in the database.		Null

SEC_PASS_POLICY

Security password policy table. Contains the default password policy and the custom password policies defined for security realms.

Dependents: SEC_IDM_REALM

Indexes: primary-key on policy_id

Field Name	Data Type		Comments	Values	Null
policy_id	number	18	ID of this record.		
pp_default	varchar2	1	Indicates whether this is the default password policy (Y/N).	D: 'N'	Null
minalpha	number	2	Minimum number of alphabetic characters for the password.		Null
minother	number	2	Minimum number of other characters (numbers or special characters) for the password.		Null
minlength	number	2	Minimum number of characters for the password. (Value is determined by either the minalpha value plus the minother value, or the minlength value, whichever is greater.)		Null
minage	number	2	Minimum age (in weeks) at which a password can be changed.		Null
maxage	number	3	Maximum age of a password. A password must be changed after a specified amount of time measured in weeks.		Null
maxexpired	number	2	Maximum number of weeks beyond the maxage value that a password can be changed by the user. After this time, the password must be reset by the security administrator.		Null
histexpire	number	2	Number of weeks that a user cannot reuse a previous password.		Null
histsize	number	2	Number of previous passwords that a user cannot reuse.		Null
mindiff	number	2	Minimum number of characters in the new password that are not in the old password. This restriction does not consider position.		Null
maxlength	number	2	Maximum number of characters for the password.		Null
maxretries	number	2	Maximum number of consecutive failed login attempts before the user account is locked.		Null
lockinterval	number	15	Time span (in minutes) during which consecutive failed login attempts are counted in determining whether to lock the user account.		Null

Field Name	Data Type		Comments	Values	Null
dictionarylist	clob		List of comma-separated words that are prohibited as passwords.		Null
minalphalower	number	2	Minimum number of alphabetic characters for the password that must be lowercase.		Null
minalphaupper	number	2	Minimum number of alphabetic characters for the password that must be uppercase.		Null

SEC_PASS_POLICY_HIST

Security password policy table. Contains the history of passwords for individual users.

Indexes: primary-key on id

Foreign Keys: (pm_user_id) must exist in SEC_IDM_USER(iu_id)

Field Name	Data Type		Comments	Values	Null
id	number	18	ID of this record.		
pm_user_id	number	18	ID for the user, from SEC_IDM_USER.iu_id.		
password	varchar2	500	Password for the user. This data is encrypted.		
create_dt	date		Date and time this record was created.		

SEC_PM_POLICY

Security policy management table. Contains data for security authorization policies. Policies are defined per security realm.

Dependents: SEC_PM_RULE_PM_POLICY

Indexes: primary-key on policy_id
unique on name

Field Name	Data Type		Comments	Values	Null
policy_id	number	19	ID of this record.		
name	varchar2	255	Name of the policy.		
description	varchar2	255	Description of the policy.		Null
realm	varchar2	255	Security realm that the policy is associated with.		
comb_alg	varchar2	255	Combining algorithm to resolve conflicts among rules in the policy.		
policy_create_dt	date		Date and time this record was created.		

SEC_PM_RULE

Security policy management table. Contains data for authorization policy rules.

Dependents: SEC_PM_RULE_PM_POLICY

Indexes: primary-key on rule_id
unique on name

Field Name	Data Type		Comments	Values	Null
rule_id	number	19	ID of this record.		
name	varchar2	255	Name of the rule.		
tr_subject	varchar2	255	Subject of the rule target.		
tr_resource	varchar2	255	Resource of the rule target.		
tr_action	varchar2	255	Action of the rule target.		
effect	varchar2	255	Effect of the rule.		
description	varchar2	255	Description of the rule.		Null
rule_create_dt	date		Date and time this record was created.		

SEC_PM_RULE_PM_POLICY

Security policy management table. Contains the association of authorization rules with authorization policies.

Indexes: primary-key on (policy_id, rule_id)

Foreign Keys: (policy_id) must exist in SEC_PM_POLICY
(rule_id) must exist in SEC_PM_RULE

Field Name	Data Type		Comments	Values	Null
policy_id	number	19	ID of the policy, from SEC_PM_POLICY.policy_id.		
rule_id	number	19	ID of the rule, from SEC_PM_RULE.rule_id.		

SEC_SESS_POLICY

Security session policy table. Contains the default session policy, created during database initialization, and the custom session policies for security realm groups.

Dependents: SEC_IDM_GROUP

Indexes: primary-key on sp_id

Field Name	Data Type		Comments	Values	Null
sp_id	number	18	ID of this record.		
soft_timer	number	15	Amount of time, in milliseconds, that a session can remain inactive (idle) before a user is automatically logged out and the session is terminated.		Null
hard_timer	number	15	Maximum duration of a session, in milliseconds, after which a user is automatically logged out and the session is terminated.		Null
sp_default	varchar2	1	Indicates whether this is the default session policy (Y/N).	D: 'N'	Null

Appendix B

Attribute Conflict-Resolution Rules

B

orded (sender
The desTir
noTiEying
ng The noTi
ieve The m
cT access To

e

v

Overview

Custom attributes can be defined at the security realm, security realm group, and user levels. Note the following points about attributes:

- All users provisioned for the realm, regardless of their group membership, inherit the realm's attributes.
- Users also inherit the attributes for all groups to which they belong.
- Individual users can also have custom attributes defined.

Because of this multiple-level attribute definition and inheritance scheme, the potential exists for conflicts among attributes. For example, suppose an attribute with the same name is defined at the realm, group, and user levels but the attribute is assigned a different value at each level. The result is an attribute conflict when determining which attribute value to use. This appendix describes the conflict-resolution rules used to resolve attribute conflicts and provides examples illustrating the rules.

Conflict-Resolution Rules

The general solution is that a user attribute overrides a group attribute, which overrides a realm attribute. Because the situation can be more complex than that general solution addresses, the following are the rules for resolving attribute conflicts:

At the *realm* level, no conflict-resolution rules exist because this is the top level.

At the *group* level, the rules are processed in the following order:

1. If a conflict exists between attributes for the priority group and a secondary group, the priority group wins.
2. If a conflict exists between attributes for secondary groups, a random selection is performed to determine which group wins. (It is expected that this conflict will be resolved at the user level. Stated another way, attributes must be defined at the user level to resolve the conflict.)
3. If a conflict exists between attributes at the realm level and group level, the group level wins. (This determination is performed last, once all group resolution rules have been performed.)

At the *user* level, if a conflict exists between attributes at the group level and the user level, the user level wins.

Examples

The examples in this section provide different scenarios to illustrate the attribute conflict-resolution rules.

Scenario 1

GROUP A (priority group defined for the USER)

Attribute: A1 = 2

GROUP B (secondary group defined for the USER)

Attribute: A1 = 4

USER

Attribute: A1 = 5

In this scenario, the USER belongs to both GROUP A and GROUP B. Each of the groups has an attribute named A1 but with different values for the attribute. The USER also has an attribute named A1, with still another value for the attribute.

Between the groups, GROUP A ($A1 = 2$) wins because it is the priority group. However, in this scenario, the user attribute ($A1 = 5$) ultimately wins because user attributes have a higher priority than group attributes.

Scenario 2

GROUP A (priority group defined for the USER)

Attribute: No A1 attribute defined

GROUP B (secondary group defined for the USER)

Attribute: $A1 = 4$

USER

Attribute: $A1 = 5$

In this scenario, the USER belongs to both GROUP A and GROUP B. The priority group has no attribute named A1, but a secondary group has an attribute named A1. The USER also has an attribute named A1, but with a different value for the attribute.

In this scenario, the user attribute ($A1 = 5$) wins because user attributes have a higher priority than group attributes.

Scenario 3

GROUP A (priority group defined for the USER)

Attribute: No A1 attribute defined

GROUP B (secondary group defined for the USER)

Attribute: $A1 = 4$

GROUP C (secondary group defined for the USER)

Attribute: $A1 = 6$

USER

Attribute: No A1 attribute defined

In this scenario, the USER belongs to GROUP A, GROUP B, and GROUP C. The priority group has no attribute named A1, and neither does the USER. Between the secondary groups with conflicting attributes named A1, the winner is determined randomly (that is, the winning attribute will be either $A1 = 4$ or $A1 = 6$).

In this scenario, the conflict must be resolved by definition of the A1 attribute and value at the user level.

Scenario 4

REALM

Attribute: $A1 = 1$

GROUP A (priority group defined for the USER)

Attribute: $A1 = 2$

GROUP B (secondary group defined for the USER)

Attribute: $A1 = 4$

USER

Attribute: A1 = 5

In this scenario, the REALM has an attribute named A1 defined. The USER belongs to both GROUP A and GROUP B, and each of the groups has an attribute named A1. The USER also has an attribute named A1. Values for the A1 attribute are different for the realm, both groups, and the user.

Between the groups, GROUP A (A1 = 2) wins because it is the priority group. However, in this scenario, the user attribute (A1 = 5) ultimately wins because user attributes have a higher priority than both group attributes and realm attributes.

Appendix C

Security Server Database Restore Operations

C

orded (send
The desti
notifying
ng The noti
ieve The m
cT access To

e

Overview

This appendix provides (1) summary information about the automated data export/database backup for the Security Server database (that is, the XE database instance) and (2) more detailed information about the manual database restore operations.

Automated Security Data Export/Database Backup

To provide a way to preserve data in the Security Server database for a fast restore, a job (`$JBOSS_HOME/jobs/db_backup.chk`) runs nightly on both nodes of the clustered UPM to perform an Oracle export dump of user data. The exported data is stored on the filesystem of both nodes. If the database has to be rebuilt from scratch, the exported data can be imported into a fresh database schema.

In addition, a backup of the Security Server database, involving Oracle's Recovery Manager (RMAN) utility, also is automated. Every night, a job runs on both nodes of the clustered UPM to perform a level 0 backup (full backup) of the Security Server database. The backup data is eventually moved from disk to tape. The backup job name is `backup_level0`. For details on this job, see the *Entity Reference*.

Methods for Restoring the Security Server Database

In the event of database corruption, two methods are available for restoring the database:

- **Preferred Method:** Use the database restore script, `$JBOSS_HOME/jobs/db_restore.chk`. This is the preferred method because it is faster. See [“Database Restore Script”](#) below.
- **Alternate Method:** Use the database restore option of the Database Backup and Restore utility to restore the database from a backup done using Oracle's RMAN utility. See [“Database Backup and Restore Utility”](#) below.

Database Restore Script

The `db_restore.chk` script, located in `$JBOSS_HOME/jobs`, relies on data exported by the `db_backup.chk` script. The restore script creates the database schema; restores data by importing the exported data into the database; and creates all constraints, synonyms, and sequences.

Database Backup and Restore Utility

**NOTE**

Information presented here about the utility is a summary, focused on restoring the Security Server database.

Restoring the Security Server database (that is, the XE database) from a backup done with Oracle's RMAN utility is a manual procedure. It is accomplished by means of the database restore option of the Database Backup and Restore utility, which is a user-interface-driven utility.

Note the following points about the database restore operation:

- A database backup must exist on disk before attempting a restore. If the backup exists only on tape, it must be brought from tape to disk first.

- Log files for Security Server database recovery operations are located in the following directory: `/backup_vol/logs/XE/backup_restore`.
- To monitor the restore operation, it is recommended that you execute the UNIX `tail` command from a separate window.

To restore the Security Server database, do the following seven steps:

1. Log in to the active UPM node as user `root`.
2. Change to the `/oracle/oracle8/backup_restore` directory and launch the user interface for the Database Backup and Recovery utility. To do this, type the following commands at the UNIX shell prompt:

```
cd /oracle/oracle8/backup_restore  
./backup_restore_ui.sh
```

The initial screen of the Database Backup and Recovery utility appears. [Figure 94](#) shows an example.

Figure 94 Database Backup and Recovery Utility — Initial Screen

Database Backup and Recovery Utility

Date: 2008-11-17 11:31:34
Host: upm1 (10.210.156.164)
ORACLE_SID:

Choose ORACLE_SID from the list below:

XE

Choose ORACLE_SID (or press ENTER to return):

3. At the prompt on the initial screen, type **XE** and press **<ENTER>**.
The utility's Main Menu screen appears. [Figure 95](#) shows an example.

Figure 95 Database Backup and Recovery Utility — Main Menu

Database Backup and Recovery Utility

Date: 2008-11-17 11:33:15
Host: upm1 (10.210.156.164)
ORACLE_SID: XE

Main Menu

1. Choose Database (ORACLE_SID)
2. Database Backup
3. Database Restore
4. Maintenance
E. Exit

Choose an option (1-4 or E):

4. At the prompt on the Main Menu screen, type **3** and press **<ENTER>**.
The Database Restore Menu screen appears. [Figure 96 on page 181](#) shows an example.

Figure 96 Database Backup and Recovery Utility — Database Restore Menu

```

Database Backup and Recovery Utility
-----
Date: 2008-11-17 11:36:42
Host: upm1 (10.210.156.164)
ORACLE_SID: XE
-----

Database Restore Menu
-----
1. Restore Database from Current Backup
2. Restore Database from Previous Backup
3. View Current Level 0 Backup Status
4. View Current Level 1 Backup Status
5. View Previous Level 0 Backup Status
6. View Previous Level 1 Backup Status

E. Return to Main Menu
-----

Choose an option (1-6 or E):

```

5. At the prompt on the Database Restore Menu screen, type a view option. (It is recommended that you view the backup status before proceeding with the database restore.) After viewing, press <ENTER> to return to the Database Restore Menu screen. You can then either choose another view option or proceed with the restore.

To proceed with the restore, at the prompt on the Database Restore Menu screen, type **1** or **2** and press <ENTER>. [If you choose 2 for previous backup, the restore procedure will swap the backup directories (current becomes previous and vice versa.)]

The Recovery Options Menu screen appears. [Figure 97](#) shows an example.

Figure 97 Database Backup and Recovery Utility — Recovery Options Menu

```

Database Backup and Recovery Utility
-----
Date: 2008-11-17 11:39:29
Host: upm1 (10.210.156.164)
ORACLE_SID: XE
-----

Recovery Options Menu
-----
1. Complete Database Recovery
2. Time-based Database Recovery
3. Tablespace Recovery
4. Datafile Recovery

E. Return to Database Restore Menu
-----

Choose an option (1-4 or E):

```

6. At the prompt on the Recovery Options Menu screen, type a recovery option and press <ENTER>.

Any option in the Recovery Options Menu restores the recovery catalog (`rcat`) database first and then proceeds with the database restore and recovery. The first two options restore the whole database, while the last two options restore one or more tablespaces or datafiles only.

Any restore procedure destroys (completely or partially) the existing database. Therefore, after you choose a recovery option, a confirmation message and prompt appear. [Figure 98 on page 182](#) shows an example.

Figure 98 Database Backup and Recovery Utility — Recovery Confirmation

```
Database Backup and Recovery utility
-----
Date: 2008-11-17 14:18:30
Host: upm1 (10.210.156.164)
ORACLE_SID: XE
-----

Complete Database Recovery
-----

You are about to restore and recover database XE
or parts of it from the CURRENT backup set.
This will destroy your existing database.
Do you want to continue (yes or no):
```

7. To proceed with the database restore, type **yes** at the confirmation prompt and press <ENTER>.

Appendix D

Cron Expressions

D

orded (sender
The destir
notifying
ng The noti
ieve The m
cT access To

e

Overview

Cron is a well-established UNIX tool. Although cron expressions are typically used for scheduling purposes (such as to schedule run times for jobs), they can also be used in determining if the current time satisfies the cron expression. Login windows in the Comverse ONE solution rely on that second type of use, and explanations in this appendix are worded accordingly. For login windows, the Security Server checks whether the current time of the login action occurs during a time that satisfies the cron expression.

Although cron expressions are powerful, they can be confusing. The purpose of this appendix is to provide enough information to enable you to create cron expressions for the loginwindow attribute, which is how login windows are defined. See [“Restricting User Logins with Login Windows” on page 29](#) for more information about this attribute.

Descriptions and Examples of Cron Expressions

A cron expression consists of six or seven fields separated by white space. [Table 15](#) describes the fields. Fields can contain any of the allowed values, along with various combinations of the allowed special characters for that field.

Table 15 Fields in Cron Expressions

Field Name	Mandatory?	Allowed Values	Allowed Special Characters
Seconds	Yes	0-59	, - * /
Minutes	Yes	0-59	, - * /
Hours	Yes	0-23	, - * /
Day of Month	Yes	1-31	, - * ? / LW
Month	Yes	1-12 or JAN-DEC	, - * /
Day of Week	Yes	1-7 or SUN-SAT	, - * ? / L #
Year	No	1970-2099	, - * /

[Table 16](#) explains the special characters and how they must be used.

Table 16 Special Characters Used in Cron Expressions

Special Character	Description
* (all values)	Used to select all values within a field. For example, * in the minute field means “every minute.”
? (no specific value)	Useful when you need to specify something in one of the two fields in which the character is allowed, but not in the other. For example, if you want to permit logins on a particular day of the month (say, the 10th), but do not care which day of the week that is, you would put 10 in the day-of-month field and ? in the day-of-week field.
- (range of values)	Used to specify ranges. For example, 10-12 in the hours field means “the hours 10, 11, and 12.”
, (additional values)	Used to specify additional values. For example, MON, WED, FRI in the day-of-week field means “the days Monday, Wednesday, and Friday.”

Table 16 Special Characters Used in Cron Expressions (Continued)

Special Character	Description
/ (increments)	Used to specify increments. For example, 0/15 in the seconds field means “the seconds 0, 15, 30, and 45,” and 5/15 in the seconds field means “the seconds 5, 20, 35, and 50.” As another example, 1/3 in the day-of-month field means “every 3 days starting on the first day of the month.” You can also specify / after the * special character. In that case, * is equivalent to having 0 before the /.
L (last)	Has different meanings in each of the two fields in which it is allowed. For example, the value L in the day-of-month field means “the last day of the month” (such as day 31 for January, and day 28 for February in years that are not leap years). If used in the day-of-week field by itself, it simply means 7 or SAT, but if used in the day-of-week field after another value, it means “the last xxxx day of the month.” For example, 6L means “the last Friday of the month.” When using the L option, it is important not to specify lists or ranges of values because you will get confusing results.
W (weekday)	Used to specify the weekday (Monday–Friday) nearest the given day. As an example, if you were to specify 15W as the value for the day-of-month field, the meaning is “the nearest weekday to the 15th of the month.” So if the 15th is a Saturday, logins will be permitted on Friday the 14th. If the 15th is a Sunday, logins will be permitted on Monday the 16th. If the 15th is a Tuesday, logins will be permitted on Tuesday the 15th. However, if you specify 1W as the value for day-of-month, and the 1st is a Saturday, logins will be permitted on Monday the 3rd because they will not “jump over” the boundary of a month's days. The W character can be specified only when the day-of-month is a single day, not a range or list of days.
# (nth)	Used to specify the “nth” xxx day of the month. For example, a value of 6#3 in the day-of-week field means “the third Friday of the month” (6 = Friday, and #3 = the third one of the month).

[Table 17](#) provides examples of cron expressions and descriptions of the examples.

**NOTE**

For login windows, the Security Server checks whether the current time of the login action occurs during a time that satisfies the cron expression. The first several rows in [Table 17](#) show examples of cron expressions for login windows. For general information, the remaining examples demonstrate how to use cron expressions for scheduling.

Table 17 Cron Expression Examples

Cron Expression	Description
Login Window Examples	
* * 8-17 * * ?	Anytime (literally, any second) between 8:00 A.M. and 5:00 P.M., every day.
* * 8-17 ? * MON-FRI	Anytime between 8:00 A.M. and 5:00 P.M., every Monday, Tuesday, Wednesday, Thursday, and Friday.

Table 17 Cron Expression Examples (Continued)

Cron Expression	Description
* 30 8-17 ? * MON-FRI	Anytime between 8:30 A.M. and 5:30 P.M., every Monday, Tuesday, Wednesday, Thursday, and Friday.
* * 12-17 * * ?	Anytime between 12:00 P.M. (noon) and 5:00 P.M., every day.
* * 0-4,20-23 * * ?	Anytime between 12:00 A.M. (midnight) and 4:00 A.M., and anytime between 8:00 P.M. and 11:00 P.M., every day.
* * 8-17 ? 3 WED 2009-2011	Anytime between 8:00 A.M. and 5:00 P.M. every Wednesday in the month of March during the years 2009, 2010, and 2011.
* * 8-17 15 * ?	Anytime between 8:00 A.M. and 5:00 P.M. on the fifteenth day of every month.
* * 8-17 L * ?	Anytime between 8:00 A.M. and 5:00 P.M. on the last day of every month.
* * 8-17 ? * 6L	Anytime between 8:00 A.M. and 5:00 P.M. on the last Friday of every month.
* * 8-17 ? * 5#3	Anytime between 8:00 A.M. and 5:00 P.M. on the third Friday of every month.
Scheduling Examples	
0 0 12 * * ?	At 12:00 P.M. (noon) every day.
0 15 10 ? * *	At 10:15 A.M. every day.
0 15 10 * * ?	At 10:15 A.M. every day.
0 15 10 * * ? *	At 10:15 A.M. every day.
0 15 10 * * ? 2009	At 10:15 A.M. every day during the year 2009.
0 * 14 * * ?	Every minute starting at 2:00 P.M. and ending at 2:59 P.M., every day.
0 0/5 14 * * ?	Every five minutes starting at 2:00 P.M. and ending at 2:55 P.M., every day.
0 0-5 14 * * ?	Every minute starting at 2:00 P.M. and ending at 2:05 P.M., every day.
0 10,44 14 ? 3 WED	At 2:10 P.M. and at 2:44 P.M. every Wednesday in the month of March.
0 15 10 ? * MON-FRI	At 10:15 A.M. every Monday, Tuesday, Wednesday, Thursday, and Friday.
0 15 10 15 * ?	At 10:15 A.M. on the fifteenth day of every month.
0 15 10 L * ?	At 10:15 A.M. on the last day of every month.
0 15 10 ? * 6L	At 10:15 A.M. on the last Friday of every month.
0 15 10 ? * 6L 2008-2011	At 10:15 A.M. on the last Friday of every month during the years 2008, 2009, 2010, and 2011.
0 15 10 ? * 5#3	At 10:15 A.M. on the third Friday of every month.
0 0 12 1/5 * ?	At 12:00 P.M. (noon) every five days, every month, starting on the first day of the month.
0 11 11 11 11 ?	Every November 11th at 11:11 A.M.

Appendix E

Well-Known Attributes

E

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

v

Overview

This appendix provides information about the well-known attributes that are part of the Comverse ONE solution. In this context, the term “well-known attributes” refers to system-defined attributes that are recognized “out of the box.” These attributes can be defined at the security realm, group, or user levels, although definition at the group level is probably the most typical.

Well-Known Attributes

[Table 18](#) describes the attributes, their default values, and the security realms where use of the attributes makes sense. See [Table 19, “Security Realms and Applications,” on page 192](#) for information on realms.

Table 18 Well-Known Attributes

Attribute Name	Default Value	Realm(s)	Description
AcctSegId	0	CSM	Identifier that groups a set of accounts. It indicates to the CSM application (Customer Center) which accounts the customer service representative (CSR) can view and work on.
BYPASS_BAL_MIN	0	CCC, CSM, CSS, SAPI	<p>Provides the ability to adjust balances below the predefined balance minimum.</p> <p>Set the BYPASS_BAL_MIN attribute to a value of 1 <i>only</i> for those users (or groups of users) who have the security permission to adjust/set the balance below the minimum value.</p> <p>The default is that no user will have the ability to set the balance below the minimum. When the attribute is not defined or set to 0, this permission will be denied.</p>
loginwindow		CSM (and any other realm where you want to control user login times)	<p>Defines when user logins are permitted. If this attribute is used, its value must be a <code>cron</code> expression. For example: loginwindow:"* * 8-17 ? * MON-FRI"</p> <p>For general information on these expressions, see Appendix D, “Cron Expressions.” Multiple <code>cron</code> expressions, separated by pipe () symbols, are supported. Enclose the expression, or group of multiple expressions, in quotation marks.</p> <p>The Hours field of the <code>cron</code> expression cannot have overlapping hours (that is, the hours cannot span into the next day). Instead, use multiple <code>cron</code> expressions to include the necessary hours.</p>
MAX_AMOUNT	100	CSM	Defines the maximum limit that CSRs can have for adjustments.

Table 18 Well-Known Attributes (Continued)

Attribute Name	Default Value	Realm(s)	Description
reseller	0	CSM, CSS, ORI, PC, SAPI	Indicates the reseller ID. A Comverse customer may delegate the “selling” of their services to other organizations. The reseller ID is used to identify sellable products (offers and bundles) that should be made visible to resellers. For example, CSM (Customer Center) users with a reseller attribute set to 10 are presented with only the offers and bundles for the reseller whose ID is 10.
usertype	system	Any realm with system users	Indicates whether the user is a system user and therefore ineligible for inactivity-related purging, as described in “Viewing Purged Inactive User Accounts,” on page 28 . This attribute is defined only at the user level. A value of <code>system</code> for the attribute identifies a system user. When the attribute is not defined, this indicates that the user is not a system user and is eligible for inactivity-related purging.

Security Realms and Applications

[Table 19](#) lists the security realms in the Comverse ONE solution and shows the applications whose users are provisioned in each realm.

Real Time only

The highlighted rows in the following table pertain to Real Time only.

Converged only

The highlighted rows in the following table pertain to Converged only.

Table 19 Security Realms and Applications

Realm Name	Application(s)
AI	Application Integrator
CCBATCH	CC Batch
CCC Real Time only	Customer Care Client
CSM Converged only	Customer and Subscriber Management GUI (Customer Center); Back Office GUI
CSS_DEALER	Realm for storing the credentials of Comverse ONE Self-Service CSR users and CSS Dealer users
CSS	Realm for the “generic” CSS user for all CSS processes and applications when no specific security token/user ID from CSS_DEALER is involved

Table 19 Security Realms and Applications (Continued)

Realm Name	Application(s)
DMROAM	Data Mediation/Roaming
ORI	Operational Reports Interface
PC	Product Catalog
REVSET	Revenue Settlements
SAPI	Single API (that is, the Unified API)
TRIVNET	Trivnet
UPSEC	Unified Platform and Security
WORKFLOW	Workflow

Appendix F

Securing CSM or Back Office Resources (CV)

F

orded (sender
The desTir
noTiEying
ng The noTi
ieve The m
cT access To

e

v

Overview

Converged only This entire appendix pertains to Converged only.

Information in this appendix is intended for Comverse personnel who deal with deployments.

This appendix provides information on how to secure resources for the CSM GUI (that is, Customer Center) and Back Office GUI. Permitting or denying access to resources establishes user permissions, based on roles. Although this discussion focuses on Customer Center, the information also generally applies to the Back Office GUI. It is specific only to these two applications.

The appendix discusses (1) the generic authorization policy spreadsheets that contain all possible securable resources and must be edited before use, (2) the types of resources that can be secured, (3) how to find the name of the particular resource you want to secure, (4) bulk loading and publishing of policies, and (5) how to verify if access to a resource is being permitted or denied.

Generic Authorization Policy Spreadsheets

Authorization policies for the Customer Center GUI and Back Office GUI contain rules that govern which subjects (that is, roles) can perform which actions on which resources in the applications. Microsoft Excel spreadsheets (also called worksheets) are used as input for bulk loading policies and their rules into the Security Server database.

Generic policy spreadsheets for the Customer Center GUI and Back Office GUI, which include all possible configurations, are available. These spreadsheets can be obtained from their respective projects in ClearCase. (For questions or assistance, contact the Front End development team.)

- **Generic CSM GUI (Customer Center) spreadsheet:** PolicyAdministration-CSM.xls
- **Generic Back Office GUI spreadsheet:** PolicyAdministration-BackOffice.xls



NOTE

A generic spreadsheet is not meant to be used as delivered. Instead, you should edit the spreadsheet based on the specific needs of your deployment. See [“Trimming Spreadsheet Rows” on page 198](#) for more details.

[Figure 99](#) shows the initial rows of the generic CSM policy spreadsheet used for the Customer Center GUI.

Figure 99 Generic CSM Policy Spreadsheet — Initial Rows

	A	B	C	D	E	
1	Node Class:	CSM				
2	Policy Realm:	CSM				
3	Policy Tag:	GUIRESOURCES				
4	Policy Description:	CSM User Policy				
5	Rule Combining Alg:	DENY-OVERRIDES ①				
6						
7	Policy Rules Begin					
8	Name	Subject	Resource	Action	Effect	Description
9	PERMIT_ALL	CSMUser	ANY ②	ANY	PERMIT	Permits All operations
10	AccountCREATE	CSMUser	SERVICE_RSRC:Account	CREATE	PERMIT	Create of object Account
11	AccountUPDATE	CSMUser	SERVICE_RSRC:Account	UPDATE	PERMIT	Update of object Account
12	AccountDELETE	CSMUser	SERVICE_RSRC:Account	DELETE	PERMIT	Delete of object Account

[Figure 99 on page 197](#) calls out two important items for special attention:

1. A value of DENY-OVERRIDES for the rule-combining algorithm means that if any given policy rule indicates to DENY an action, that will override any more generic PERMIT rule.
2. A value of ANY for a resource or action is a special term. It is interpreted by the Security Server as meaning any possible resource or action in the application.

Each authorization rule in the policy appears in a row of the spreadsheet, starting with row 9. The columns related to rules are as follows:

- **Name:** Name of the rule.
- **Subject:** Name of a security role. The role must exist in the Security Server database before a policy is used.
- **Resource:** Name of a resource that can be secured. For details about the different types of resources, see [“Types of Resources” on page 199](#).
- **Action:** The action on the resource that is permitted or denied by the rule, based on the value in the Effect column. For details, see [“Valid Actions Based on Resource Type” on page 200](#).
- **Effect:** Effect of the rule, which is either to permit or deny the action on the given resource.
- **Description:** General description of the rule.

Trimming Spreadsheet Rows

[Figure 100](#) shows more rule rows in the generic CSM policy spreadsheet. The purpose of this spreadsheet is to include all possible securable resources for the application. You must trim the number of rows in the spreadsheet to a reasonable set.

Figure 100 Generic CSM Policy Spreadsheet — More Rows

	A	B	C	D	E	F
1	Node Class:	CSM				
2	Policy Realm:	CSM				
3	Policy Tag:	GUIRESOURCES				
4	Policy Description:	CSM User Policy				
5	Rule Combining Alg:	DENY-OVERRIDES				
6						
7	Policy Rules Begin					
8	Name	Subject	Resource	Action	Effect	Description
9	PERMIT_ALL	CSMUser	ANY	ANY	PERMIT	Permits All operations
10	AccountCREATE	CSMUser	SERVICE_RSRC.Account	CREATE	PERMIT	Create of object Account
11	AccountUPDATE	CSMUser	SERVICE_RSRC.Account	UPDATE	PERMIT	Update of object Account
12	AccountDELETE	CSMUser	SERVICE_RSRC.Account	DELETE	PERMIT	Delete of object Account
13	Account.AccountCategoryWRITE	CSMUser	ATTRIBUTE_RSRC.Account.AccountCategory	WRITE	PERMIT	Writing to attribute Account.AccountCategory
14	Account.AccountExternalIdWRITE	CSMUser	ATTRIBUTE_RSRC.Account.AccountExternalId	WRITE	PERMIT	Writing to attribute Account.AccountExternalId
15	Account.AccountExternalIdTypeWRITE	CSMUser	ATTRIBUTE_RSRC.Account.AccountExternalIdType	WRITE	PERMIT	Writing to attribute Account.AccountExternalIdType
16	Account.AccountInternalIdWRITE	CSMUser	ATTRIBUTE_RSRC.Account.AccountInternalId	WRITE	PERMIT	Writing to attribute Account.AccountInternalId
17	Account.AccountRatingStatusWRITE	CSMUser	ATTRIBUTE_RSRC.Account.AccountRatingStatus	WRITE	PERMIT	Writing to attribute Account.AccountRatingStatus
18	Account.AccountStatusWRITE	CSMUser	ATTRIBUTE_RSRC.Account.AccountStatus	WRITE	PERMIT	Writing to attribute Account.AccountStatus
19	Account.AccountStatusDtWRITE	CSMUser	ATTRIBUTE_RSRC.Account.AccountStatusDt	WRITE	PERMIT	Writing to attribute Account.AccountStatusDt
20	Account.AccountTypeWRITE	CSMUser	ATTRIBUTE_RSRC.Account.AccountType	WRITE	PERMIT	Writing to attribute Account.AccountType
21	Account.AcctSegIdWRITE	CSMUser	ATTRIBUTE_RSRC.Account.AcctSegId	WRITE	PERMIT	Writing to attribute Account.AcctSegId



CAUTION

The generic CSM policy spreadsheet contains several thousand rows. The Security Server can handle only several hundred rows being loaded into a policy. If the rows are not trimmed to a reasonable set, they will overload the memory of the Security Server.

The goal of trimming rows in the spreadsheet is to select which minimal number of PERMIT rows to keep based on the needs of your particular deployment. For example, for any given subject (that is, role), you normally would keep the row for the PERMIT_ALL rule, which permits ANY access to ANY resource. Then, to deny access to specific resources, you would keep the rows for those resources but change the value in the Effect column to DENY. (With DENY-OVERRIDES as the

policy's rule-combining algorithm, a value of DENY overrides the PERMIT permission of the PERMIT_ALL rule.) You would then delete all other rule rows for that subject from the spreadsheet. See [“Types of Resources”](#) below for more information about the resources.



NOTE

The generic CSM policy spreadsheet includes only one subject (that is, role), which is CSMUser. This is a default role created during the initial Security Server database installation/configuration. In a production environment, you might or might not be using that role name. If not, change it to a role suitable for your deployment. Also, an application in a production environment typically requires multiple roles, and rules need to be provided for those roles. You can include multiple comma-separated roles in the Subject column of the spreadsheet, assuming the permission in the Action column is valid for each role. For different permissions, copy the initially supplied row for CSMUser, change the value in the Subject column to a different role, and change the permission in the Action column. The goal is to trim the rows for each role to the fewest number of rows that meet the needs of the deployment.

Types of Resources

The following three types of resources can be secured: SERVICE_RSRC, ATTRIBUTE_RSRC, and ACTION_RSRC.

- **SERVICE_RSRC:** SERVICE_RSRC rows are method calls to the Unified API (also known as the Single API or SAPI). [Figure 101](#) shows an example of SERVICE_RSRC rows in the generic CSM policy spreadsheet. In one of these rows, a value of DENY in the Effect column ensures that this method is not allowed to be called for a CREATE, UPDATE, or DELETE operation on an application Domain Object (such as Account). *Typically, denying access to a SERVICE_RSRC is a last resort.*

The reason this is a last resort is that it does not produce a user-friendly error. Denying access to a SERVICE_RSRC stops the behavior past the point where the application can present a pertinent error message. Better options are (1) to secure a panel (ACTION_RSRC) so that the user cannot get into a screen that would allow them to do the operation or (2) to secure attributes (ATTRIBUTE_RSRC) that gray out fields on the screen. Denying access to a SERVICE_RSRC could be a second layer of security. For example, you could secure a panel to provide a good user experience but also secure the service just to be certain that an UPDATE, for instance, could never be called on the object.

- **ATTRIBUTE_RSRC:** ATTRIBUTE_RSRC rows are Domain Object attributes (that is, GUI fields). [Figure 101](#) shows an example of ATTRIBUTE_RSRC rows in the generic CSM policy spreadsheet. In one of these rows, a value of DENY in the Effect column results in the GUI showing any field bound to the attribute as read-only (that is, grayed out).

Figure 101 SERVICE_RSRC and ATTRIBUTE_RSRC Resources

AccountCREATE	CSMUser	SERVICE_RSRC:Account	CREATE	PERMIT
AccountUPDATE	CSMUser	SERVICE_RSRC:Account	UPDATE	PERMIT
AccountDELETE	CSMUser	SERVICE_RSRC:Account	DELETE	PERMIT
Account.AccountCategoryWRITE	CSMUser	ATTRIBUTE_RSRC:Account.AccountCategory	WRITE	PERMIT
Account.AccountExternalIdWRITE	CSMUser	ATTRIBUTE_RSRC:Account.AccountExternalId	WRITE	PERMIT
Account.AccountExternalIdTypeWRITE	CSMUser	ATTRIBUTE_RSRC:Account.AccountExternalIdType	WRITE	PERMIT

- **ACTION_RSRC:** Using an ACTION_RSRC is the preferred way of securing a feature. ACTION_RSRC rows map to GUI behaviors and are derived from the ApplicationConfig.xml file, which is an application file that defines actions for the GUI. [Figure 102](#) shows an example of ACTION_RSRC rows in the generic CSM policy spreadsheet. Each action in ApplicationConfig.xml maps to an ACTION_RSRC row in the spreadsheet. In the GUI, these actions are commonly invoked by menu, navigation bar, button, or hyperlink clicks.

Figure 102 ACTION_RSRC Resources

rate-discount-dlgREAD	CSMUser	ACTION_RSRC:rate-discount-dlg	READ
rate-discount-dynamic-dlgREAD	CSMUser	ACTION_RSRC:rate-discount-dynamic-dlg	READ
rate-unit-cr-dlgREAD	CSMUser	ACTION_RSRC:rate-unit-cr-dlg	READ
rate-unitcr-dynamic-dlgREAD	CSMUser	ACTION_RSRC:rate-unitcr-dynamic-dlg	READ
adjustment-invoice-search-dlgREAD	CSMUser	ACTION_RSRC:adjustment-invoice-search-dlg	READ
paymentdistribution-invoice-search-dlgREAD	CSMUser	ACTION_RSRC:paymentdistribution-invoice-search-dlg	READ

Valid Actions Based on Resource Type

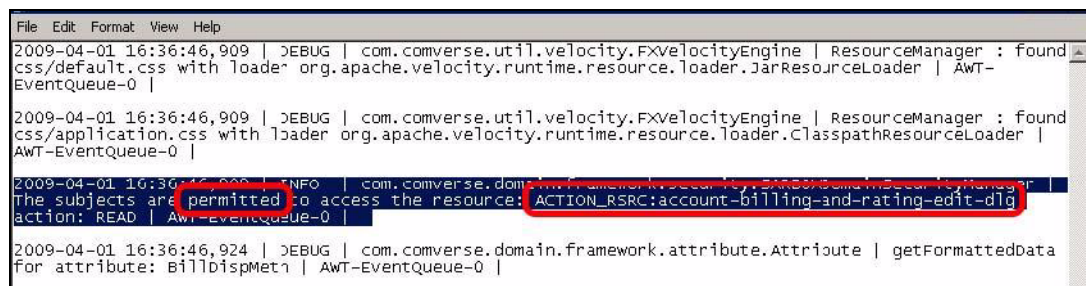
[Table 20](#) describes the valid actions (permissions) for the Action column, based on the type of resource involved.

Table 20 Valid Actions by Resource Type

Type of Resource	Description	Valid Actions
SERVICE_RSRC	Resources for Domain Objects	CREATE, UPDATE, DELETE
ATTRIBUTE_RSRC	Resources for Domain Object attributes (GUI fields)	WRITE (DENY for this type of resource results in the GUI field being read-only, displayed as grayed out on the screen.)
ACTION_RSRC	Resources for actions in the GUI (such as a new panel, button, wizard, navigation bar link)	READ (PERMIT or DENY for this type of resource determines whether the user can navigate/invoke the action through the GUI.)

Finding a Resource to Secure

Resources to secure can be located by opening the application's log file (CSM.log for Customer Center) with a text editor such as Notepad and reviewing the log. [Figure 103](#) shows an example of the CSM.log file, with an ACTION_RSRC highlighted. The log file is located on the PC in the directory where the application is installed.

Figure 103 CSM Log File

An alternate and easier method, if you have access to a Windows `tail` utility, is to run that utility for the log file while using the application. The utility enables you to see changes to the log scrolling by in real-time.

In the log file or while tailing the log, note which resources get requested when you perform operations that you want to secure. To deny access to a resource, look up that same resource name in the spreadsheet and change the Effect to DENY.

Bulk Loading and Publishing the Policy

After editing the generic policy spreadsheet appropriately, use the bulk load operation to load the authorization policy and its rules into the Security Server database. See [“Bulk Policy Operations” on page 45](#), starting with step 3, for information.

After the policy has been loaded into the database, it must be published before the application can use it. For details, see [“Publishing Policies” on page 43](#).



NOTE

Resynchronizing policies is not necessary for the Customer Center and Back Office GUI applications. They automatically retrieve their published policies at startup after a successful user login.

Verifying the Policy

In the Customer Center and Back Office GUI authorization policies, a role is used as the subject in authorization rules. Before verifying your policy, be aware of the related requirements. These requirements must be met before a policy can control an individual user's permissions in the application, based on a role. The following list provides a summary:

- The user must be provisioned in the CSM security realm in the Security Server database. (This is the realm for both Customer Center and Back Office GUI users.)
- An appropriate role (role that was used as the subject in the policy's rules) must exist in the Security Server database.
- The role must be associated with a group in the CSM security realm.
- The user must belong to the group that has the role associated with it. (A user inherits a role from a group.)
- The authorization policy must be published, which makes it available for the application to retrieve.
- The user must log in to the application after the policy is published.

To verify that your policy is working as expected, log in as a user who belongs to the appropriate group in the CSM security realm. When the application starts up, it retrieves the policy and starts using the policy's authorization rules.

To determine whether the authorization policy is securing a resource as intended, review the log file (`CSM.log` for Customer Center). The file resides in the directory on the PC where the application is installed. Use the application and navigate or invoke the action you intended to secure. In the log file, the text reads “subjects are permitted” if the authorization policy allowed the action. The text reads “subjects are denied” if the policy prohibited the action. The highlighted action in [Figure 103 on page 200](#) shows an example.

Appendix G

Securing Security GUI Resources

G

orded (sender
The desti
notifying
ng The noti
ieve The m
cT access To

e

v

Overview

Information in this appendix is intended for Comverse personnel who deal with deployments.

This appendix provides information on how to secure resources for the Web-based Security graphical user interface (that is, the Security GUI). Permitting or denying access to resources establishes user permissions, based on roles. Stated another way, permitting or denying access to resources controls what a user can see and do in the GUI.

The appendix discusses (1) the default authorization policy spreadsheet that must be edited before use; (2) how resources align with GUI elements, indicating how to find the particular resource you want to secure; (3) a summary of bulk loading and publishing of policies; and (4) how to verify if the policy rules are working as expected.

Default Authorization Policy Spreadsheet

The authorization policy for the Security GUI contains rules that govern which subjects (that is, roles) can perform which actions on which resources in the application. A Microsoft Excel spreadsheet (also called a worksheet) is used as input for bulk loading the policy and its rules into the Security Server (XE) database.

The location/name of the default Security GUI policy spreadsheet is as follows:

- `$JBOSS_HOME/batch/policy/PolicyAdministration_UP_SECGUI_new.xls`

This default spreadsheet has the default tags for resources delivered. The spreadsheet can be edited, based on the deployment's customized security requirements.

The default spreadsheet provides the following:

- **ADMIN role:** Has access to all resources and can perform any action in the Security GUI.
- **GUEST role:** Can view all resources (that is, can view all areas of the GUI), but cannot add, edit, or delete information.



NOTE

Edit the spreadsheet based on the specific needs of your deployment. (Make a backup copy and store it elsewhere before editing.) See [“Editing Spreadsheet Rows” on page 208](#) for more details.

After the spreadsheet is edited appropriately, place it in the following location for bulk loading:

- `$JBOSS_HOME/batch/policy`

In the spreadsheet, be aware of the following information:

1. A value of PERMIT-OVERRIDES for the rule-combining algorithm means that if any given policy rule indicates to PERMIT an action, that will override any more generic DENY rule.
2. A value of ANY for a subject, resource, or action is a special term. It is interpreted by the Security Server as meaning any possible subject, resource, or action.

Each authorization rule in the policy appears in a row of the spreadsheet, starting immediately after the row labeled “Policy Rules Begin.” The columns related to rules are as follows:

- **Name:** Name of the rule.
- **Subject:** Name of a security role. Multiple comma-separated roles are supported, assuming the permissions are the same for each role. The role(s) must exist in the Security Server database (XE) before a policy is used.

- **Resource:** Name of a resource that can be secured. For information about the resources, see [“Types of Resources and their Alignment with the Security GUI.”](#)
- **Action:** The action on the resource that is permitted or denied by the rule, based on the value in the Effect column. Multiple comma-separated actions are supported. For more information, see [“Types of Actions” on page 208.](#)
The spreadsheet includes all possible actions for each resource.
- **Effect:** Effect of the rule, which is either to permit or deny the action on the given resource.
- **Description:** General description of the rule.

Types of Resources and their Alignment with the Security GUI

The Security GUI follows a hierarchical order for resources. The resources themselves are self-explanatory and include the name of the perspective and the tab that appear in the GUI. The perspective refers to the top-level tab in the GUI, and the tab refers to the second-level tab that appears on a top-level tab.

[Figure 104](#) shows some representative rows in the spreadsheet that illustrate this concept. For example, the `^Identity$` resource refers to the top-level Identity tab in the Security GUI, the `^Identity.users$` resource refers to the Users tab on that top-level Identity tab, the `^Identity.groups$` resource refers to the Groups tab on that top-level Identity tab, and so on.

Figure 104 Default Security GUI Policy Spreadsheet — Representative Rows

Name	Subject	Resource	Action	Effect	Description
SECGUI_PERMIT_IDENTITY	ADMIN,GUEST	^Identity\$	View	PERMIT	Permit Identity tab
SECGUI_PERMIT_IDENTITY_USERS	ADMIN	^Identity.users\$	Add,Edit,Delete	PERMIT	Allow
SECGUI_PERMIT_IDENTITY_USERS_ATTRIBUTES	ADMIN	Identity.users.attributes	Add,Edit,Delete	PERMIT	Allow
SECGUI_PERMIT_IDENTITY_GROUPS	ADMIN	^Identity.groups\$	Add,Edit,Delete	PERMIT	Allow
SECGUI_PERMIT_IDENTITY_GROUPS_ATTRIBUTES	ADMIN	Identity.groups.attributes	Add,Edit,Delete	PERMIT	Allow
SECGUI_PERMIT_IDENTITY_ROLES	ADMIN	^Identity.roles\$	Add,Edit,Delete	PERMIT	Allow
SECGUI_PERMIT_IDENTITY_REALMS	ADMIN	^Identity.realms\$	Add,Edit,Delete	PERMIT	Allow
SECGUI_PERMIT_IDENTITY_REALMS_ATTRIBUTES	ADMIN	Identity.realms.attributes	Add,Edit,Delete	PERMIT	Allow
SECGUI_PERMIT_IDENTITY_REALMS_PASSWORD_POLICY	ADMIN	Identity.realms.passwordpolicy	Edit	PERMIT	Allow



NOTE

The `^` character at the beginning and the `$` character at the end of a resource (such as `^Identity$`) are part of Perl pattern-matching syntax. They are used to instruct the Security Server to match a rule only with the complete word.

As an example of how permissions for actions on resources affect the GUI, see [Figure 105 on page 207](#) and [Figure 106 on page 207](#). The two figures show how the GUI looks when users with different roles are logged in to the Security GUI.

In [Figure 105](#), the logged-in user belongs to a security group that is associated with the ADMIN role. The ADMIN role has permission for all actions (View, Add, Edit, and Delete) on the ^Identity.users\$ resource. ([Figure 104 on page 206](#) shows these permissions in the spreadsheet.)

Figure 105 Security GUI — All Permissions for the ^Identity.users\$ Resource

IDENTITY KEY POLICY AUDIT CREDENTIAL											
Users Groups Roles Realms											
Search User											
User Name	First Name	Last Name	Realm	Email	Lock	Department	Phone	Force ChangePassword	LastUpdated	Reset Password	Operation
Guest	Guest	Guest	UPSEC	guest@comverse.com	No			No	07/30/2010 6:19:08 PM	Reset Password	✗
aparna	Aparna	Reddy	UPSEC	aparna@test.com	No			No	07/21/2010 3:18:35 PM	Reset Password	✗
secadmin	Admin_FN	Admin_LN	UPSEC	--	No	dept001	(856) 608-7407	No	07/30/2010 6:28:54 PM	Reset Password	✗
test234	test	test234	UPSEC	test@tst.com	No			Yes	07/29/2010 4:28:24 PM	Reset Password	✗
user3	user33333	user3	UPSEC	user3@comverse.com	No			No	07/10/2010 7:48:15 AM	Reset Password	✗
user5	User5	User5	UPSEC	vfgghf@hygj.fg	No	fgfghf	76789678678	No	06/30/2010 4:40:03 AM	Reset Password	✗
userAdmin	User	Admin	UPSEC	userAdmin@comverse.com	No	ytry	656	No	07/30/2010 9:48:16 AM	Reset Password	✗
userSec	User	Sec	UPSEC	user@comverse.com	No	rteter	5345	No	07/30/2010 4:12:08 PM	Reset Password	✗
											Add User

In [Figure 106](#), the logged-in user belongs to a security group that is associated with the GUEST role. The GUEST role has permission only for the View action (no permission for Add, Edit, or Delete) on the ^Identity.users\$ resource. ([Figure 104 on page 206](#) shows these permissions in the spreadsheet.) This user can only view information and cannot add, edit, or delete any information.

Figure 106 Security GUI — Only View Permission for the ^Identity.users\$ Resource

IDENTITY KEY POLICY AUDIT CREDENTIAL										
Users Groups Roles Realms										
Search User										
User Name	First Name	Last Name	Realm	Email	Lock	Department	Phone	Force ChangePassword	LastUpdated	
Guest	Guest	Guest	UPSEC	guest@comverse.com	No			No	07/30/2010 6:19:08 PM	
aparna	Aparna	Reddy	UPSEC	aparna@test.com	No			No	07/21/2010 3:18:35 PM	
secadmin	Admin_FN	Admin_LN	UPSEC	--	No	dept001	(856) 608-7407	No	07/30/2010 5:45:58 PM	
test234	test	test234	UPSEC	test@tst.com	No			Yes	07/29/2010 4:28:24 PM	
user3	user33333	user3	UPSEC	user3@comverse.com	No			No	07/10/2010 7:48:15 AM	
user5	User5	User5	UPSEC	vfgghf@hygj.fg	No	fgfghf	76789678678	No	06/30/2010 4:40:03 AM	
userAdmin	User	Admin	UPSEC	userAdmin@comverse.com	No	ytry	656	No	07/30/2010 9:48:16 AM	
userSec	User	Sec	UPSEC	user@comverse.com	No	rteter	5345	No	07/30/2010 4:12:08 PM	

Perspective (Top-Level Tab) Access Control

To manage access for users at the perspective level, the resource name corresponds to the perspective name (that is, the name of the top-level tab in the Security GUI).

Table 21 Resources for Perspectives (Top-Level Tabs) in the Security GUI

Resource	Top-Level Tab
^Identity\$	Identity
^Key\$	Key

Table 21 Resources for Perspectives (Top-Level Tabs) in the Security GUI (Continued)

Resource	Top-Level Tab
^Policy\$	Policy
^Audit\$	Audit
^Credential\$	Credential

If a role does not have at least View permission at the perspective level, users with that role will not even see the corresponding top-level tab in the GUI.

Tab-Level Access Control

To manage access for users at the tab level, the resource name corresponds to the perspective (top-level tab in the GUI) and the tab (second-level tab, one level down on the top-level tab). For example, as mentioned previously, the ^Identity.users\$ resource (see [Figure 104 on page 206](#)) corresponds to the Users tab on the top-level Identity tab.

If a role does not have at least View permission at the tab level, users with that role will not even see the corresponding tab in the GUI.

Additional Access Control

Depending on the particular area in the Security GUI, additional resources provide further access control. For example, this is the case for the Identity.users.attributes resource shown in [Figure 104 on page 206](#). This resource controls the ability to add, edit, and delete user attributes in the Security GUI.

Types of Actions

Four kinds of actions are used in the Security GUI and are included in the Action column in the Security GUI spreadsheet:

- **Add:** To add a new element of the resource type.
- **Edit:** To modify an existing resource element.
- **Delete:** To delete an existing resource element.
- **View:** To view the resource.

The Security GUI spreadsheet includes all the valid actions for each resource.

Editing Spreadsheet Rows

You must edit the rows in the spreadsheet. Refer to the default spreadsheet for all possible securable resources.

Based on the needs of your deployment, add new role names as needed for the various resources. You can copy rows for existing rules and add them to the spreadsheet, changing the role names to those used by your deployment. You can also add the new role names to the Subject column in existing rows. (Multiple comma-separated roles are permitted in the Subject column, assuming the permissions are valid for all the roles.)



CAUTION

The Security Server can handle only several hundred rows being loaded into a policy. If the rows are not kept to a reasonable number, they will overload the memory of the Security Server.

Bulk Loading and Publishing the Policy

After editing the policy spreadsheet appropriately, use the bulk load operation to load the authorization policy and its rules into the Security Server (XE) database. See [“Bulk Policy Operations” on page 45](#), starting with step 3, for information.

After the policy has been loaded into the database, it must be published and resynchronized. For details, see [“Publishing Policies” on page 43](#) and [“Resynchronizing Policies” on page 44](#).

Verifying the Policy

Before verifying your policy, be aware of the related requirements. These requirements must be met before a policy can control an individual user’s permissions in the Security GUI, based on a role. The following list provides a summary:

- The user must be provisioned in the UPSEC security realm in the Security Server (XE) database.
- An appropriate role (role that was used as the subject in the policy’s rules) must exist in the Security Server database.
- The role must be associated with a group in the UPSEC security realm.
- The user must belong to the group that has the role associated with it. (A user inherits a role from a group.)
- The authorization policy must be published and resynchronized, which makes it available for use.
- The user must log in to the application after the policy is published and resynchronized.

To verify that your policy is working as expected, log in as a user who belongs to the appropriate group in the UPSEC security realm. When the Security GUI starts up, it starts using the policy’s authorization rules. Check to make sure that the user’s ability to view and work with information is how it should be, based on the policy’s rules.

Index

A

- AAA 3, 7
- AcctSegId attribute 191
- Advanced Encryption Standard 11
- AES 11
- attributes, conflict-resolution rules 173
- attributes, well-known 191
- audit event index 53
- audit management 10, 49
 - and accountability 10
 - API 10
 - definition 10
 - framework 10
- audit records 10
 - format 51
 - purging 51
 - querying 50
- auditable events 10
- auditing, enabling/disabling 50
- authentication
 - definition 8
 - relationship to identity management 8
 - Web services interface 8
- authentication, authorization, and accounting 3, 7
- authorization
 - and policy management 9
 - definition 9
- authorization policy
 - creating 42
 - creating advanced policy manually 46
 - creating with bulk load operation 45
 - deleting 43
 - modifying 42
 - publishing 43
 - resynchronizing 44
 - viewing 41
- authorization rule 38
 - creating 39
 - creating with bulk load operation 45
 - deleting 40
 - modifying 39

B

- Blowfish 11
- bulk load
 - authorization policies and rules 45
 - database passwords 70
 - user accounts and related entities 31
- BYPASS_BAL_MIN attribute 191

C

- checkpointing, of audit records 10

- command line interface
 - batch mode 16, 37, 64
 - interactive mode 15, 37, 49, 64
 - noninteractive mode 15, 37, 49, 64
- credentials
 - creating 67
 - deleting 68
 - modifying 68
 - publishing database credentials 69
 - viewing 66
- credentials management 64
 - API 11
 - framework 10
 - operations 66
- cron expressions 29, 185
- CSM and Back Office GUI resources, securing 197

D

- data encryption 63
 - API 11
 - framework 10
- database passwords, *See* credentials
- default password policy 18, 24
- DEFAULT security realm group 19
- Distributed Audit Service 10, 49

E

- encryption algorithms 11
- encryption key management 7, 64
 - operations, symmetric keys 65
 - See also* symmetric encryption key
- Excel worksheet
 - for bulk load of authorization policies and rules 45
 - for bulk load of database passwords 70
 - for bulk load of user accounts/related entities 31
- Extensible Access Control Markup Language 9

G

- GKID 63
- group, *See* security realm group

H

- hard timeout 17, 19, 120

I

- identity life cycle 7, 15

- identity management 7, 15
 - administrator interface 8
 - API 8
 - custom repositories 8
 - definition 7, 15
 - framework 8
 - object-relational mapping (ORM) API 8
 - perspectives 15
 - relationship to authentication 8
 - Security Server repository 8
- IDMAdministrationTemplate.xls 31

L

- LDAP 8
- Lightweight Directory Access Protocol 8
- locking user account 26
- logins, restricting with login windows 29
- loginwindow attribute 29, 191

M

- Management Shell security-related CLI commands 119
 - add_group 120
 - add_realm 120
 - add_role 121
 - add_user 122
 - build_report 133
 - change_password 123
 - create_auth_policy 129
 - create_auth_rule 130
 - create_key 134
 - delete_key 134
 - disable_key 134
 - disable_user 123
 - enable_key 134
 - enable_user 123
 - find_users 123
 - global options for 119
 - list_attributes 123
 - list_auth_policy 130
 - list_auth_rule 131
 - list_credential 135
 - list_groups 124
 - list_keys 135
 - list_purged_users 124
 - list_realms 124
 - list_roles 124
 - list_users 125
 - lock_user 125
 - modify_auth_policy 131
 - modify_auth_rule 131
 - modify_group 125
 - modify_realm 126
 - modify_user 127
 - publish_credential 135

- publish_policy 132
- remove_auth_policy 132
- remove_auth_rule 132
- remove_credential 136
- remove_group 128
- remove_realm 128
- remove_role 128
- remove_user 129
- reset_password 129
- resync_policy 133
- store_credential 136
- unlock_user 129
- MAX_AMOUNT attribute 191
- mshell 15, 37, 49, 64, 119

O

- OASIS 9, 35
- object-relational mapping API 8
- Organization for the Advancement of Structured Information Standards 9, 35
- ORM API 8

P

- PAP 9, 35, 37
- password policies, understanding 24
- password policy
 - default 18, 24
 - reverting to the default policy 20, 126
 - security realm, creating 18
 - security realm, modifying 20
 - security realm, viewing 124
- password, database, *See* credentials
- password, user
 - changing your own password 30
 - resetting user passwords 30
- passwords, dictionary list prohibiting passwords 18
- Payment Card Industry Data Security Standard 7, 63
- PCI DSS 7, 63
- PDP 9, 35, 36
- PEP 9, 35, 36
- Policy Administration Point 9, 35, 37
- Policy Decision Point 9, 35, 36
- Policy Enforcement Point 9, 35, 36
- policy management 9, 35
 - and authorization 9
 - API 9
 - definition 9
 - framework 9
 - See also* authorization policy
- priority group 17, 25
- purged inactive user accounts
 - account state determining purge 29
 - viewing 28

R

- RBAC 9
- realm group, and security roles 17
- realm group, *See* security realm group
- realm, *See* security realm
- reseller attribute 192
- resources, securing for Customer Center and Back Office GUI 197
- Rivest, Shamir, Adleman 11
- role, *See* security role
- Role-Based Access Control 9
- RSA 11
- rule, *See* authorization rule

S

- SAML 8
- Sarbanes-Oxley Act of 2002 3, 7
- SARBOX 3, 7, 49
- Security Assertion Markup Language 8
- Security GUI
 - logging in 75
 - logging out 77
- security platform 7
 - services 7
 - summary of features 7
- security realm
 - creating 17
 - creating with bulk load operation 31
 - custom attributes 17
 - definition 17
 - deleting 21
 - modifying 20
- security realm group
 - creating 19
 - creating with bulk load operation 31
 - custom attributes 19
 - DEFAULT 17
 - definition 17
 - deleting 21
 - modifying 20
 - priority group 25
- security role 9
 - creating 23
 - creating with bulk load operation 31
 - default roles 17
 - definition 22
 - deleting 23
 - modifying 23
- Security Server API 8, 10
- Security Server database
 - automated backup 179
 - automated data export 179
 - methods to restore 179

Security Server database tables

- SEC_AA_EVENT 144
- SEC_DPM_PASSWORD 146
- SEC_IDM_COUNTER 147
- SEC_IDM_GROUP 148
- SEC_IDM_GROUP_ATTRIBUTE 149
- SEC_IDM_GROUP_ROLE 150
- SEC_IDM_PURGED_USERS 151
- SEC_IDM_REALM 152
- SEC_IDM_REALM_ATTRIBUTE 153
- SEC_IDM_ROLE 154
- SEC_IDM_USER 155
- SEC_IDM_USER_ATTRIBUTE 157
- SEC_IDM_USER_ROLE 158
- SEC_IDM_USER_SESSION 159
- SEC_KM_KEYS 160
- SEC_KM_SERVER_KEYS 161
- SEC_NETWORK_DEV_CRED 162
- SEC_PASS_POLICY 163
- SEC_PASS_POLICY_HIST 165
- SEC_PM_POLICY 166
- SEC_PM_RULE 167
- SEC_PM_RULE_PM_POLICY 168
- SEC_SESS_POLICY 169

security solution overview 7

session timeouts, defaults 19, 120

soft timeout 17, 19, 120

SOX, *See* SARBOX

standard interfaces

- SAML 8
- XACML 9
- XDAS 10

symmetric encryption key

- creating 65
- deleting 66
- disabling 65
- enabling 66
- viewing 65

T

The Open Group 49

timeout, session

defaults 19, 120

hard timeout for realm group 17, 19, 120

soft timeout for realm group 17, 19, 120

U

Unified Platform 3

Unified Platform Agent 9, 10, 43

unlocking user account 26

UPA 9, 10, 44

upgrades, and database/OS user passwords 71

- user account
 - creating 25
 - creating with bulk load operation 31
 - custom attributes 26
 - deleting 28
 - enabling/disabling for inactivity purge 29
 - finding across security realms 24
 - locking/unlocking 26
 - modifying 27
 - resetting password 30
 - viewing 25
 - viewing all custom attributes 25
 - viewing purged inactive accounts 28
- user password
 - changing your own 30
 - resetting user passwords 30

W

- Web services interface 8
- well-known attributes 191
 - loginwindow 191

X

- XACML 9, 35
- XACML policy 35
- XDAS 10, 49, 51
 - event codes 53
 - event outcome codes 57