

Novell Kerberos KDC

1.0.3

www.novell.com

ADMINISTRATION GUIDE

June 27, 2007



Novell[®]

Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export, or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See www.novell.com/info/exports/ for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Copyright © 2005-2007 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

Novell, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed on the Novell Legal Patents Web page (<http://www.novell.com/company/legal/patents/>) and one or more additional patents or pending patent applications in the U.S. and in other countries.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.

www.novell.com

Novell Kerberos KDC Administration Guide
[June 05, 2007](#)

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

For Novell trademarks, see the Novell Trademark and Service Mark list (<http://www.novell.com/company/legal/trademarks/tmlist.html>).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Copyright © 1985-2002 by the Massachusetts Institute of Technology. Export of software employing encryption from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Furthermore if you modify this software you must label your software as modified software and not distribute it in such a fashion that it might be confused with the original MIT software. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

The implementation of the Yarrow pseudo-random number generator in `src/lib/crypto/yarrow` has the following copyright:

Copyright 2000 by Zero-Knowledge Systems, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Zero-Knowledge Systems, Inc. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Zero-Knowledge Systems, Inc. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

ZERO-KNOWLEDGE SYSTEMS, INC. DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL ZERO-KNOWLEDGE SYSTEMS, INC. BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTUOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Kerberos V5 includes documentation and software developed at the University of California at Berkeley, which includes this copyright notice:

Copyright © 1983 Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Berkeley and its contributors. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notices and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided also that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions.

Contents

	About This Guide	9
1	Overview	11
	Overview of Kerberos	11
	Commonly Used Kerberos Terminology	11
	How Does Kerberos Work	12
	Understanding Novell Kerberos KDC	12
	Novell Kerberos KDC Features	13
	Novell Kerberos KDC Components	13
	Changes to the MIT KDC Code Base	14
2	Kerberos KDC Integration with eDirectory	17
	Understanding Kerberos Integration with eDirectory	17
	Kerberos Container	19
	Realm Container	20
	Principal Objects	21
	Kerberos Service Objects	22
	Ticket Policy Object	22
	Password Policy Object	23
	Administrative Considerations While Integrating Kerberos with eDirectory	23
3	Managing Novell Kerberos KDC	25
	The krb5.conf Configuration File	25
	Configuration and Administration Utilities	26
	The kdb5_util Utility	27
	The kadmin Utility	28
	Managing Realms	30
	Creating a Realm	30
	Modifying a Realm	32
	Viewing a Realm	35
	Destroying a Realm	35
	Listing Realms	36
	Managing Services	36
	Creating a Service	37
	Modifying a Service	38
	Viewing a Service	39
	Listing Services	39
	Destroying a Service	40
	Setting a Password for Service Objects	41
	Setting the Server Certificate	42
	Managing Principals	43
	Adding a Principal	43
	Modifying a Principal	47
	Deleting a Principal	47
	Listing Principals	48
	Getting Principal Information	48

Setting Principal Password	49
Extracting Principal Key to a Keytab File	50
Removing Keytab Entry	50
Managing Ticket Policies	51
Creating a Ticket Policy	51
Modifying a Ticket Policy.	53
Destroying a Ticket Policy	53
Viewing a Ticket Policy	54
Listing Ticket Policies	55
Managing Password Policies	55
Adding a Password Policy	56
Modifying a Password Policy	56
Deleting a Password Policy	57
Viewing Policy Values	58
Listing Policies	58
Updating Kerberos LDAP Extension Information	58
Importing Trusted Root Certificate.	59
Setting the Master Key	60
Changing Principal Password	60
4 Integrating Universal Password	61
Configuring Universal Passwords	61
Prerequisites	61
Integrating Universal Password with Novell Kerberos KDC	61
Kerberos Password Agent.	63
Universal Password Considerations.	64
5 Deployment Notes	67
Optimizing the Performance.	67
LDAP Connection Pool	67
Configuring LDAP Connection Pool	68
Bulkloading Principals	69
6 Interoperability with MIT and Microsoft KDCs	71
Interoperability with MIT KDC	71
Accessing Services in mitrealm from novrealm	71
Accessing Services in novrealm from mitrealm	71
Interoperability with Microsoft KDC	72
Accessing Services in w2kdomain from novrealm	72
How Cross-Realm Setup Works.	74
7 Security Considerations	75
8 Troubleshooting Kerberos KDC	77
Installation	77
Starting the Services	77
KDC	78
kdb5_util.	78
kadmin.	80
Password Agent	80
iManager Plug-in	81
A Sample krb5.conf File	83
B Supported Encryption Types and Salt Types	85
Supported Encryption Types	85
Supported Salt Types	85

C Administrative Privileges for Kerberos Database

87

About This Guide

This guide describes Novell® Kerberos KDC and provides information on how to administer it.

The guide is intended for Novell eDirectory™ or Kerberos administrators and is divided into the following sections:

- ◆ Chapter 1, “Overview,” on page 11
- ◆ Chapter 2, “Kerberos KDC Integration with eDirectory,” on page 17
- ◆ Chapter 3, “Managing Novell Kerberos KDC,” on page 25
- ◆ Chapter 4, “Integrating Universal Password,” on page 61
- ◆ Chapter 5, “Deployment Notes,” on page 67
- ◆ Chapter 6, “Interoperability with MIT and Microsoft KDCs,” on page 71
- ◆ Chapter 7, “Security Considerations,” on page 75
- ◆ Chapter 8, “Troubleshooting Kerberos KDC,” on page 77
- ◆ Appendix A, “Sample krb5.conf File,” on page 83
- ◆ Appendix B, “Supported Encryption Types and Salt Types,” on page 85
- ◆ Appendix C, “Administrative Privileges for Kerberos Database,” on page 87

Documentation Updates

You can find the latest version of this documentation at the [Novell Documentation Website \(http://www.novell.com/documentation/kdc/index.html\)](http://www.novell.com/documentation/kdc/index.html).

Additional Documentation

- ◆ [Novell eDirectory 8.7.3 Documentation \(http://www.novell.com/documentation/edir873/index.html\)](http://www.novell.com/documentation/edir873/index.html)
- ◆ [Novell eDirectory 8.8 Documentation \(http://www.novell.com/documentation/beta/edir88/index.html\)](http://www.novell.com/documentation/beta/edir88/index.html)
- ◆ [Kerberos Documentation \(http://web.mit.edu/kerberos/www/\)](http://web.mit.edu/kerberos/www/)

Documentation Conventions

In Novell documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

When a single pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname is presented with a backslash. Users of platforms that require a forward slash, such as Linux* or UNIX*, should use forward slashes as required by your software.

User Comments

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comment feature at the bottom of each page of the online documentation, or go to www.novell.com/documentation/feedback.html and enter your comments there.

1

Overview

This chapter introduces Kerberos and Novell Kerberos KDC.

- ♦ [“Overview of Kerberos” on page 11](#)
- ♦ [“Understanding Novell Kerberos KDC” on page 12](#)

Overview of Kerberos

Kerberos is a standard protocol that provides a means of authenticating entities on a network and is based on a trusted third-party model. It involves shared secrets and uses symmetric key cryptography. Kerberos was developed at the Massachusetts Institute of Technology (MIT).

MIT created Kerberos as a solution to network security problems. The Kerberos protocol uses strong cryptography so that a client can prove its identity to a server (and vice versa) across an insecure network connection. After a client and server have used Kerberos to prove their identity, they can also encrypt all of their communication to assure privacy and data integrity.

Kerberos is a solution to your network security problems. It provides the tools of authentication and strong cryptography over the network to help you secure your information systems across your entire enterprise.

This chapter introduces you to Kerberos and its concepts:

- ♦ [“Commonly Used Kerberos Terminology” on page 11](#)
- ♦ [“How Does Kerberos Work” on page 12](#)

Commonly Used Kerberos Terminology

The following table lists the definitions of some commonly used Kerberos terminologies.

Table 1 Kerberos Terminologies

Terminology	Definition
Key (also referred to as Secret Key)	Encryption key shared by a principal and the KDC, distributed outside the system, with a long lifetime. In the case of a user's principal, the key is derived from a password.
Principal	Entity in the network. Each entity corresponds to a principal.
Realm	Logical grouping of principals.
Service	Resource provided to network clients, such as mail server.
Session key	Temporary encryption key used between two principals, with a lifetime limited to the duration of a single login “session”.

Terminology	Definition
Service ticket	Required to access services in the network.
Ticket	Record that helps a client authenticate itself to a server. It contains information such as client's identity, a session key, a timestamp, and other information—all sealed using the server's secret key.
Ticket Granting Ticket (TGT)	Initial ticket obtained after a successful login. This ticket is used to get the service ticket to access a service.

How Does Kerberos Work

Kerberos uses the concept of a central server called the Key Distribution Center (KDC). The KDC contains the identities and keys of every principal in the network that must service within its realm. This principal information is stored in a local database within the KDC. In Novell® Kerberos KDC, the principal and realm information is stored in Novell eDirectory™

A typical KDC provides the following basic services:

- ◆ **Authentication Server (AS):** Issues authentication credentials known as Ticket Granting Tickets (TGT) to users while logging in.
- ◆ **Ticket Granting Server (TGS):** Issues service tickets to the users in response to their requests accompanied by TGT so that they can access various services in the realm.

Kerberos provides the following additional services and utilities to manage KDC and Kerberos principals:

- ◆ **Kerberos Administration Server:** Server component for maintaining Kerberos principals, policies, and service key tables (keytabs). This server responds to the requests from the `kadmin` and `kpasswd` utilities.
- ◆ **Kerberos Administration Utilities:** Client component (such as `kadmin`, `kadmin.local`, and `kdb5_util`) for maintaining Kerberos realms, principals, policies, and service key tables.
- ◆ **Kerberos Password Server:** Server component of the Kerberos Password utility for changing passwords of Kerberos principals.
- ◆ **Kerberos Client Utilities:** Utilities such as `kinit` and `kpasswd`, which are used for various operations like login and changing passwords.

For more information on the Kerberos solution developed by the MIT, refer to the *Kerberos System Administrator's Guide* (<http://web.mit.edu/kerberos/www/>).

Understanding Novell Kerberos KDC

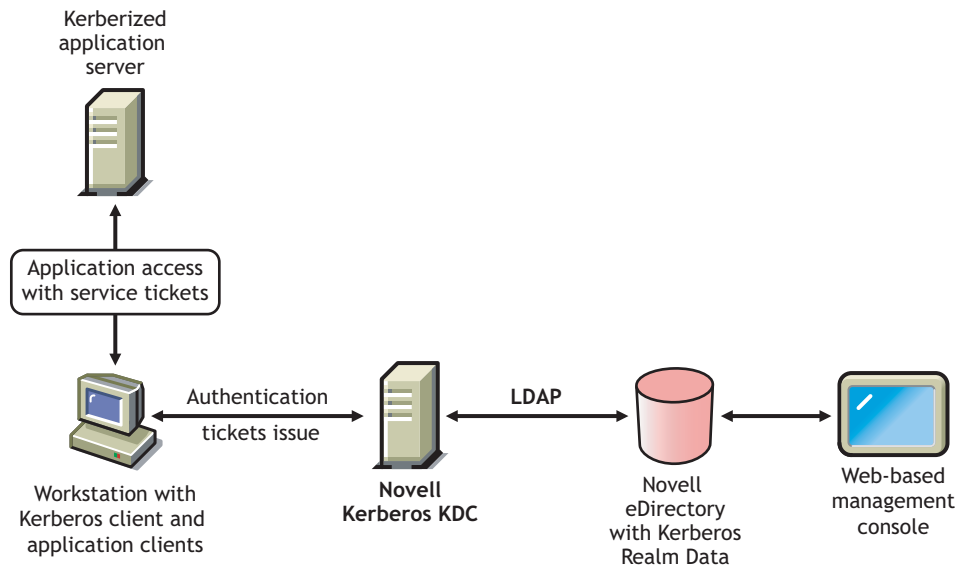
Traditional Kerberos implementations store relevant Kerberos information pertaining to a realm in a database. Database propagation between KDCs are handled by vendor-specific protocols. The Kerberos database is managed using vendor-specific administration utilities.

Novell® Kerberos KDC provides the ease of single point of management for deployments with both Kerberos and Novell eDirectory™, and gives the advantage of eDirectory replication and security capabilities. It moves Kerberos-specific data to eDirectory and provides Kerberos services using a KDC that accesses data stored in eDirectory. Additionally, because authentication requests lead to database operations that are mostly read-only in nature, eDirectory is well suited to replace the traditional database component.

Novell Kerberos KDC integrates Kerberos Authentication, Administration, and Password Servers with eDirectory as data store. Administration is possible both using the traditional command line tools and Novell's Web-based framework, iManager.

Novell Kerberos KDC is derived from [MIT implementation of Kerberos \(http://web.mit.edu/kerberos\)](http://web.mit.edu/kerberos). It is interoperable with the Kerberos implementations from other vendors like Microsoft* (Active Directory).

Figure 1 Kerberos Authentication Using Novell Kerberos KDC



This chapter provides information about the following:

- ◆ “Novell Kerberos KDC Features” on page 13
- ◆ “Novell Kerberos KDC Components” on page 13
- ◆ “Changes to the MIT KDC Code Base” on page 14

Novell Kerberos KDC Features

Novell Kerberos KDC provides the following features:

- ◆ A standard authentication method to leverage your existing eDirectory deployment.
- ◆ An iManager interface to manage multiple Kerberos realms.
- ◆ Universal password integration that enables you to use the same password to login to both eDirectory and KDC.

Novell Kerberos KDC Components

This section introduces you to the components of Novell Kerberos KDC.

- ◆ “Key Distribution Center (KDC)” on page 14
- ◆ “Kerberos Administration Server” on page 14
- ◆ “Kerberos Password Server” on page 14

- ♦ “kdb5_util and kadmin” on page 14
- ♦ “Kerberos LDAP Extensions” on page 14
- ♦ “Kerberos Password Agent” on page 14

Key Distribution Center (KDC)

KDC provides authentication and ticket granting services to Kerberos clients. The principal and realm information is stored in eDirectory. Novell Kerberos KDC accesses this information using secure LDAP connections.

Kerberos Administration Server

The Administration server services administrative requests such as principal management and key tab operations. This server acts like any another kerberized service on the network and requires the corresponding service ticket to perform any operation.

Kerberos Password Server

The Password server provides the necessary functionality to change principals' passwords from standard Kerberos Change Password clients. Users who want to use this service to change their passwords need to authenticate to KDC first and get the service ticket for this Password server. Though the wire-level protocol for this change password is still not a standard, this server complies with the Internet Draft on the Kerberos Change Password Protocol.

kdb5_util and kadmin

kdb5_util and kadmin are tools for managing the Kerberos Realm and principals in eDirectory. For more information on these utilities refer to [Chapter 3, “Managing Novell Kerberos KDC,” on page 25](#).

Kerberos LDAP Extensions

Kerberos LDAP extensions service the requests for storing and retrieving various Kerberos-specific keys from eDirectory, for example, the master key of a Realm. The keys are stored in eDirectory in a secure form.

Kerberos Password Agent

Kerberos Password Agent keeps the Kerberos password in sync with the universal password. Therefore, it needs to be deployed when universal password integration is required. It synchronizes the Kerberos password with universal password whenever the universal password is set in eDirectory.

Changes to the MIT KDC Code Base

- ♦ Allows usage of eDirectory (through LDAP) as a database.
- ♦ Allows usage of OpenSSL over NCI for cryptographic operations.
- ♦ Tight integration of Kerberos and eDirectory identities, including a single password by means of universal password.
- ♦ Separate Password server instead of the Administration server playing that role.

- ◆ Modifications to kadmin for eDirectory integration. kadmin.local works with the LDAP server and not on the local database.
- ◆ Modifications to kdb5_util to work with eDirectory.
- ◆ Additions to the krb5.conf configuration file to include eDirectory configuration.
- ◆ Multithreading of KDC and implementation of the LDAP connection pool mechanism to improve performance, reliability, and scalability.

2

Kerberos KDC Integration with eDirectory

Novell® Kerberos KDC, provides the ease of single point of management for deployments with both Kerberos and Novell eDirectory™, and gives the advantage of eDirectory replication and security capabilities. It moves Kerberos-specific data to eDirectory and provides Kerberos services using a KDC that accesses data stored in eDirectory.

Understanding Kerberos Integration with eDirectory

In a Kerberos system, the entities in a network are called principals and a logical grouping of principals is called a realm.

In Novell Kerberos KDC, the realms and principals of Kerberos are mapped to eDirectory as shown in the following table:

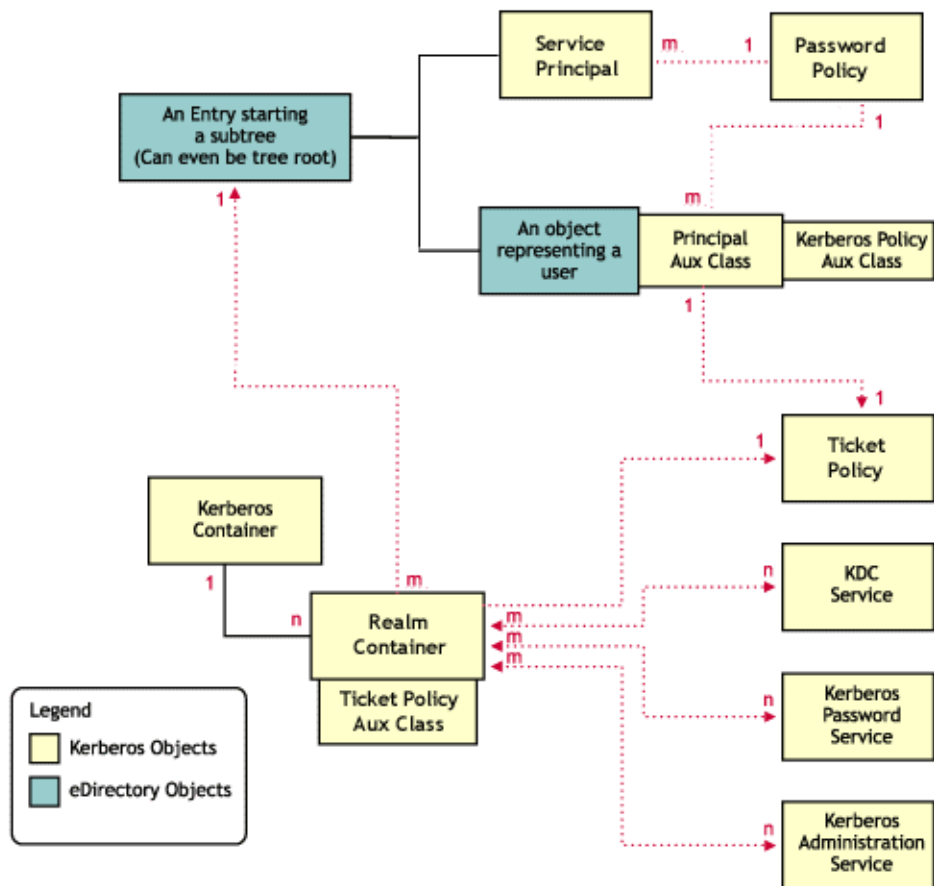
Table 2 Kerberos Mapping With eDirectory

Kerberos Terms	Mapping to eDirectory
Realm	<p>Can be mapped to:</p> <ul style="list-style-type: none"> ♦ a sub-tree or a container. ♦ an entire eDirectory tree. It can be the [Root] too. <p>For example, if eDirectory has a container, HR, you can create a realm, HRREALM that references to the HR container. All the users in the container HR belong to the realm, HRREALM.</p>
Principal	<p>Can be mapped to a user object or to a service object; called user principal and service principal respectively.</p> <p>For example, if an eDirectory tree has FTP as a service object and John as an user object, you can add</p> <ul style="list-style-type: none"> ♦ a user principal, John. ♦ a service principal, FTP. <p>Kerberos specific object class krbPrincipalAux get added to the objects.</p>

You can create realms in eDirectory and add principals to these realms. You can associate these realms and principals to one or more eDirectory users. For information on creating realms, adding principals and managing them, refer to [Chapter 3, “Managing Novell Kerberos KDC,” on page 25](#).

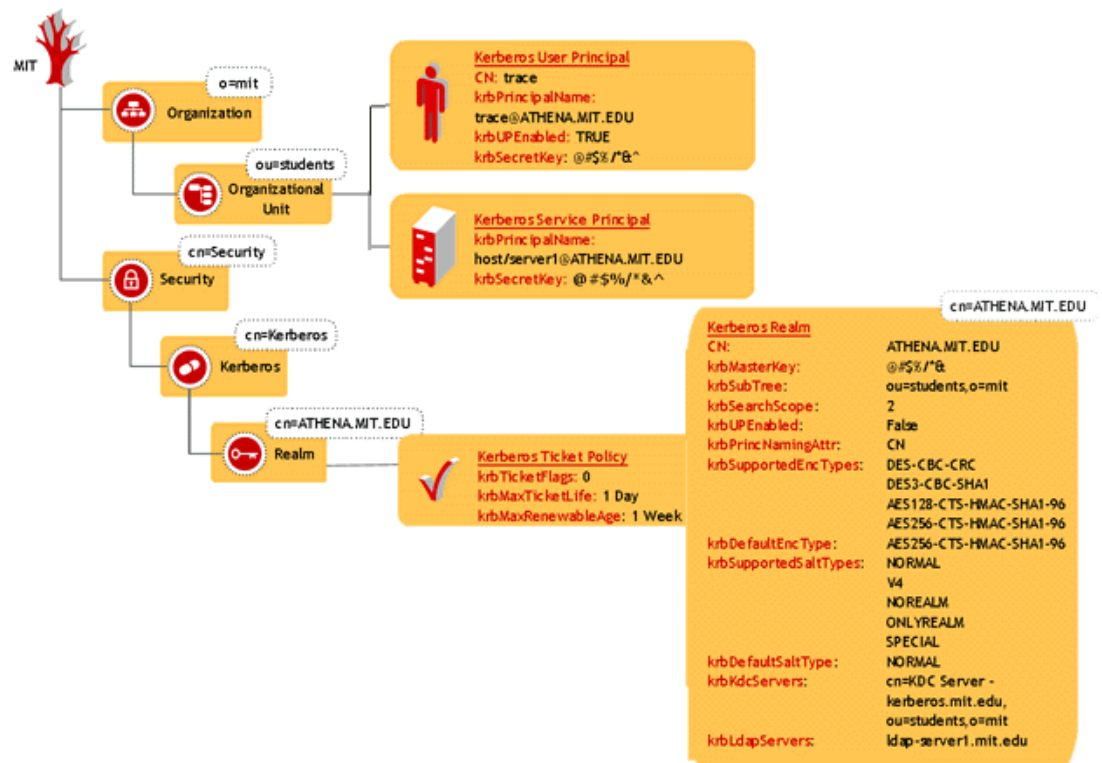
You need to create the realms under the Kerberos container that can be located anywhere in the eDirectory tree. This helps you to administer the Kerberos objects easily.

Figure 2 Kerberos Integration with eDirectory



The following diagram illustrates how the Kerberos data is mapped in eDirectory:

Figure 3 eDirectory and Kerberos Mapping



This section provides information on:

- ◆ “Kerberos Container” on page 19
- ◆ “Realm Container” on page 20
- ◆ “Principal Objects” on page 21
- ◆ “Kerberos Service Objects” on page 22
- ◆ “Ticket Policy Object” on page 22
- ◆ “Password Policy Object” on page 23
- ◆ “Administrative Considerations While Integrating Kerberos with eDirectory” on page 23

Kerberos Container

Kerberos container contains only the realm objects. All the realm objects in the tree are placed in this container. This makes the Kerberos administration easier. The Kerberos container can be located anywhere in the tree but the location of the container is stored in the security container.

The Kerberos container can be

- ◆ In the security container under the root (default); or
- ◆ Under any sub-tree or container

IMPORTANT: Make sure that Kerberos container is replicated on all eDirectory servers that Kerberos services (KDC, Administration, and Password services) are configured with.

Kerberos Container Reference Attributes

The following table describes the Kerberos container attribute:

Table 3 Kerberos Container Attributes

Attribute	Description
Container name	Name of the container object.

Security Container Reference Attributes

Security Container contains an attribute that gives the location of the Kerberos container.

Table 4 Security Container Attributes

Attribute	Description
Container reference	DN of the Kerberos container.

Realm Container

The realm container stores the realm name and related realm information for Kerberos authentication and administration, password management servers to process requests. This object contains krbtgt, kadmin/admin, kadmin/changepw, kadmin/history, internal principals, and any other service principal.

Realm Container Attributes

The following table describes the realm container attributes:

Table 5 Description of the Realm Container Attributes

Attribute	Description
Realm name	Name of the realm. This is unique within an eDirectory tree.
Supported encryption types	List of all the encryption types supported by the realm.
Supported salt types	List of all the salt types supported by the realm.
Default encryption type	The default encryption type supported by the realm.
Default salt type	The default salt type supported by the realm.
Master key	Realm specific master key.
Search scope	Scope for searching the principals under the specified subtree.
UP enabled	Specifies whether to use the universal password of the user as the Kerberos password or not.

Realm Container Associations

The following table describes the objects and servers you can associate the realm container to:

Table 6 Realm Container Associations

Associate to	Description
Policy reference	Reference to a policy object. All the principals belonging to the realm container.
Subtree	Reference to an entry that starts a sub-tree under which the principals of the realm are placed.
KDC servers	List of references to the KDC service objects that can service the realm.
Administration servers	List of references to the administration service objects that can service the realm.
Password servers	List of references to the password service objects that can service the realm.

Principal Objects

A Principal is a fundamental entity in Kerberos. All the services, clients, and users are represented as principals in Kerberos. Principals are associated with keys.

Principal Attributes

The following table describes the principal attributes:

Table 7 Principal Attributes

Attributes	Description
Principal name	Name of the principal. This is used to uniquely identify a principal within a realm.
Principal expiration	This is the time at which the principal expires.
Principal (secret) key	This is a set of all the secret keys that are associated with a principal. The version, type and other information about the keys are stored in this attribute.
UP enabled	This attribute specifies whether to use the universal password of the user as the Kerberos password or not.

Principal Associations

The following table describes the object you can associate a principal to:

Table 8 Principal Associations

Associate to	Description
Ticket policy	Reference to a ticket policy object that is applicable to a particular principal.
Password policy	Reference to a Kerberos password policy object that is applicable to a particular principal.

Kerberos Service Objects

This represents the Kerberos services namely KDC, Administration, and Password services.

Each service of Novell Kerberos KDC (KDC server, Administration server, and Password server) uses a representative object in eDirectory. This has two purposes:

- ◆ To treat the service as a client of eDirectory and provide necessary authorization
- ◆ To store any configuration related to the service

When each service comes up, it makes an LDAP bind to eDirectory as the corresponding service object, using the stashed password or stored certificate on the local system. All subsequent operations happen based on the rights provided to that object.

Kerberos Service Attributes

The following table describes the Kerberos service attributes:

Table 9 Kerberos Service Attributes

Attributes	Description
Host server	This attribute holds the host name, transport protocol and port for a Kerberos service.

Kerberos Service Associations

The following table describes the object you can associate a Kerberos service to:

Table 10 Kerberos Service Associations

Associate to	Description
Realm references	List of references to the realm objects.

Ticket Policy Object

This is a Kerberos ticket policy object. This object can be located anywhere in the eDirectory tree.

Ticket Policy Attributes

The following table describes the policy attributes:

Table 11 Policy Attributes

Attributes	Description
Ticket flags	Various ticket flags that can be allowed for a principal.
Maximum ticket lifetime	Maximum lifetime of a ticket for a principal in seconds.
Maximum ticket renewable lifetime	Maximum lifetime for a principal's ticket in seconds.

Password Policy Object

This is a Kerberos password policy object. This object can be located anywhere in the eDirectory tree.

Password Policy Attributes

The following table describes the password policy attributes:

Table 12 Password Policy Attributes

Attributes	Description
Maximum password life	Maximum lifetime of a principal's password.
Minimum password life	Minimum lifetime of a principal's password.
Password minimum characters	Minimum number of character classes allowed in a password.
Password minimum length	Minimum length of the password.
Password history length	Number of previous versions of passwords that are stored.
Password policy count	Number of principals that refer to this policy.

Administrative Considerations While Integrating Kerberos with eDirectory

Administrators are advised to consider the following points while deploying Novell Kerberos KDC, as it involves an integration between Kerberos and eDirectory paradigms:

- ◆ Multiple realms can be configured in a single tree.
- ◆ Multiple Kerberos identities can be associated with a single user identity. This follows the preferred way of managing eDirectory data, that of storing all data pertaining to the user object on itself.
- ◆ Overall, the need for more than one administrator is avoided. The benefits of having such a single point of management are ease of administration and reduced administrative costs.
- ◆ In case separate eDirectory container administrators being present, each will have an additional responsibility of administrating the Kerberos data.

3

Managing Novell Kerberos KDC

This chapter provides the following information on managing Novell Kerberos KDC:

- ◆ “The krb5.conf Configuration File” on page 25
- ◆ “Configuration and Administration Utilities” on page 26
- ◆ “Managing Realms” on page 30
- ◆ “Managing Services” on page 36
- ◆ “Managing Principals” on page 43
- ◆ “Managing Ticket Policies” on page 51
- ◆ “Managing Password Policies” on page 55
- ◆ “Updating Kerberos LDAP Extension Information” on page 58
- ◆ “Importing Trusted Root Certificate” on page 59
- ◆ “Setting the Master Key” on page 60
- ◆ “Changing Principal Password” on page 60

The krb5.conf Configuration File

You can use the `/etc/krb5.conf` configuration file to set the default values. While managing Novell Kerberos KDC, when you do not specify any of the mandatory parameters, the values are taken from the `/etc/krb5.conf` file.

For a sample configuration file, refer to [Appendix A, “Sample krb5.conf File,” on page 83](#).

Table 13 krb5.conf Configuration File Details

Parameter	Description
libdefaults	
<code>default_realm</code>	Default name of the realm.
realms	
<code>max_life</code>	Specifies the maximum life-time of ticket issued.
<code>max_renewable_life</code>	Specifies the maximum life-time to which issued ticket can be renewed.
<code>acl_file</code>	File name and path of the ACL file.
<code>dict_file</code>	File name and path of the DICT file.

Parameter	Description
kdc	KDC server name.
admin_server	Administration server name.
kpasswd_server	Password server name.
database_module	Database module configuration tag (reference to the one used in 'dbmodules' section.)
kdcdefaults	
num_threads	Number of threads to be used by KDC, Administration server, or Password server.
domain_realm	Domain-realm mappings.
logging	
kdc	File name and path of the KDC log file.
admin_server	File name and path of the Administration server log file.
kpasswd_server	File name and path of the Password server log file.
dbdefaults	
database_module	Database module configuration tag (reference to the one used in 'dbmodules' section.)
dbmodules	
db_library	Library name.
ldap_ssl_port	LDAP port number.
ldap_kdc_dn	KDC service object DN.
ldap_kadmind_dn	Administration service object DN.
ldap_kpasswd_dn	Password service object DN.
ldap_root_certificate_file	Path of trusted root certificate file.
ldap_service_password_file	Path of the service stashed file.
realm_read_refresh_interval	Interval (in seconds) at which realm configuration needs to be re-read by KDC, Administration server, or Password server.
ldap_servers	List of LDAP servers.
ldap_conns_per_server	Number of LDAP connections to be used by KDC, Administration server, or Password server.

Configuration and Administration Utilities

The **kdb5_util** utility helps you manage realms, Kerberos services, and ticket policies.

The **kadmin** utility helps you manage principals, password policies, and keytab entries.

You can also use iManager to configure and administer Novell Kerberos KDC.

The kdb5_util Utility

The syntax is as follows:

```
kdb5_util [-D user_dn [-w passwd]] [-h ldap_server] [-p ldap_port]
[-t trusted_cert] cmd [cmd_options]
```

The kdb5_util parameters are described below:

Table 14 kdb5_util Parameter Description

Parameter	Description
-D	Distinguished name of the user who has sufficient rights to authenticate to the LDAP server.
-w	Userdn password. We do not recommend you to use this option as the password is visible when you enter it through command line.
-h	Hostname or IP address of the server hosting the LDAP service for a Kerberos realm.
-p	SSL port number of the LDAP server.
-t	Specifies the filename that contains Trusted Root Certificate of the LDAP server.

The command options include the following:

Table 15 kdb5_util Command Options

Command Options	Description
create	Creates a realm.
modify	Modifies a realm.
view	Displays the attributes of a realm.
destroy	Destroys a realm.
list	Lists all the realms.
create_service	Creates a KDC, Administration, or Password service.
modify_service	Modifies a KDC, Administration, or Password service.
view_service	Displays the service details.
destroy_service	Deletes the service.
list_service	Lists all the services.
create_policy	Creates a ticket policy.
modify_policy	Modifies a ticket policy.
view_policy	Displays the ticket policy details.

Command Options	Description
<code>destroy_policy</code>	Deletes a ticket policy.
<code>list_policy</code>	Lists all the ticket policies.
<code>setsrvpw</code>	Set a password for the service objects such as KDC, Administration, and Password server in eDirectory.
<code>setsrvcert</code>	Configures the service to use the issued certificate for authentication instead of the password.
<code>import_cert</code>	Imports the trusted root certificate from eDirectory.
<code>ldapxtn_info</code>	Updates the ldapExtensionInfo attribute on the LDAP server object.
<code>setmasterkey</code>	Sets the master key password.

The kadmin Utility

You can use the `kadmin` or `kadmin.local` utilities to manage principals, keys, and password policies. In Novell Kerberos KDC, `kadmin.local` is used to access the database (eDirectory) remotely, unlike MIT Kerberos.

`kadmin` is a client utility and contacts the Administration server, which in turn contacts eDirectory for any administration request.

`kadmin.local` directly contacts eDirectory for completing the administration request.

The syntax is as follows:

```
kadmin [-r realm] [-p principal] [-q query] [-s admin_server[:port]]
[-w password] [[-c ccache]|[-k [-t keytab]]]
```

```
kadmin.local [-r realm] [-p principal] [-q query] [-x db_args] [-d dbname] [-e
"enc:salt ..."] [-m]
```

```
cmd [cmd_options]
```

The `kadmin` and `kadmin.local` parameters are described below:

Table 16 `kadmin` and `kadmin.local` Parameter Description

Parameter	Description
<code>-r</code>	Specifies the Kerberos realm. By default, the <code>default_realm</code> parameter of the <code>krb.conf</code> file is used.
<code>-p</code>	Specifies the principal you will authenticate to.
<code>-q</code>	Passes query directly to <code>kadmin</code> , which will perform query and then exit.
<code>-s</code>	Specifies the admin server which <code>kadmin</code> should contact.
<code>-c</code>	Specifies to use <code>credentials_cache</code> as the credentials cache. The <code>credentials_cache</code> should contain a service ticket for the <code>kadmin/admin</code> service; it can be acquired with the <code>kinit(1)</code> program. If this option is not specified, <code>kadmin</code> requests a new service ticket from the KDC, and stores it in its own temporary <code>ccache</code> .

Parameter	Description
-k	Uses a keytab to decrypt the KDC response instead of prompting for a password on the keyboard. In this case, the default principal will be host/hostname. If there is not a keytab specified with the t option, then the default keytab will be used.
-t	Uses keytab to decrypt the KDC response. This can only be used with the -k option.
-x	Specifies database-specific parameters. <ul style="list-style-type: none"> ◆ -x nconns=<number_of_connections> Same as the ldap_conns_per_server parameter in the configuration file. ◆ -x port=<port_number> Same as the ldap_ssl_port parameter in the configuration file. ◆ -x host=<hostname> Same as the ldap_servers parameter in the configuration file.. This option is a multivalued option. ◆ -x binddn=<bind_dn> Equates to ldap_kdc_dn, ldap_kadmind_dn depending on the services that is being invoked. For example, if the service is KDC, then binddn equates to ldap_kdc_dn ◆ -x bindpwd=<bind_password> There is no corresponding option in the conf file. This option overrides the password that will read from the ldap_service_password_file. ◆ -x cert=<certificate_file> Same as ldap_root_certificate_file parameter from the conf file. This option is a multivalued option. ◆ -x dbname=<database_name> Specifies the name of the Kerberos database. This is applicable only while using the local database (DB2) as the database backend and not LDAP. <<rephrase>>
-d	Specifies the name of the Kerberos database.
-e	Sets the list of encryption types and salt types to be used for any new keys created. NOTE: If universal password integration is enabled, refer to "Key Generation" on page 63 .
-m	Do not authenticate using a keytab. This option will cause kadmin to prompt for the master database password.
-w	Uses the password specified and does not prompt for it. NOTE: Placing the password for a Kerberos principal with administration access into a shell script can be dangerous if unauthorized users get read access to the script.

The command options include the following:

Table 17 kadmin and kadmin.local Command Options

Command Options	Description
add_principal , addprinc , ank	Adds a principal.

Command Options	Description
<code>delete_principal, delprinc</code>	Deletes a principal.
<code>modify_principal, modprinc</code>	Modifies a principal.
<code>change_password, cpw</code>	Sets the principal password.
<code>get_principal, getprinc</code>	Displays the attributes of a principal.
<code>list_principals, listprincs, get_principals, getprincs</code>	Lists all the principals.
<code>add_policy, addpol</code>	Adds a password policy.
<code>modify_policy, modpol</code>	Modifies a password policy.
<code>delete_policy, delpol</code>	Deletes a password policy.
<code>get_policy, getpol</code>	Displays the attributes of a password policy.
<code>list_policies, listpols, get_policies, getpols</code>	Lists the password policies.
<code>ktadd, xst</code>	Adds entries to a keytab.
<code>ktremove, ktrem</code>	Removes entries from a keytab.

The `-x db_args` specifies the following database-specific parameters:

Managing Realms

You can manage realms by using the `kdb5_util` utility.

This section provides information about the following:

- ◆ “Creating a Realm” on page 30
- ◆ “Modifying a Realm” on page 32
- ◆ “Viewing a Realm” on page 35
- ◆ “Destroying a Realm” on page 35
- ◆ “Listing Realms” on page 36

Creating a Realm

You can create realm using either of the following methods:

- ◆ [Command Line \(page 30\)](#)
- ◆ [iManager \(page 32\)](#)

Command Line

Use the following syntax to create a realm:

```
kdb5_util [-D user_dn] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]

create [-subtree subtree_dn] [-sscope search_scope]
       [-ldapdn ldap_server_list] [-kdcn kdc_service_list]
```

```

[-admindn admin_service_list] [-pwddn passwd_service_list]
[-enctypes supported_enc_types] [-defenctype default_enc_type]
[-salttypes supported_salt_types] [-defsalttype default_salt_type]
[-policy policy_dn] [-up]
[-k mkeytype] [-m|-P password|-f stashfilename]
[-r realm]

```

For example:

```

kdb5_util -r ATHENA.MIT.EDU -D cn=admin,o=org -h ldap-server1.mit.edu create
-sscope 2 -kcdn cn=service-kdc,o=org:cn=service-kdc2,o=org -enctypes des-
cbc-crc:des3-cbc-shal -defenctype des3-cbc-shal -salttypes normal:onlyrealm
-defsalttype normal -policy cn=rpolicy,o=org

```

Output of the above command:

```

Password for "cn=admin,o=org":
Initializing database for realm 'ATHENA.MIT.EDU'
Enter KDC database master key:
Re-enter KDC database master key to verify:


```

Table 18 Create Realm Parameter Description

Parameter	Description
-subtree	Subtree where principals and other Kerberos objects in the realm are placed.
-sscope	Scope for searching the principals under the specified subtree. The parameter sscope specifies the search scope for searching the principals under the subtree specified. The possible values are 1 or one (one level), 2 or sub (subtree).
-ldapdn	List of LDAP servers that the Kerberos servers (KDC and administration servers) can contact. The list contains the DNs of the LDAP servers separated by a colon (:).
-kcdn	List of KDC Service objects serving the realm. The list contains the DNs of the KDC Service objects separated by a colon (:).
-admindn	List of Administration Service objects serving the realm. The list contains the DNs of the Administration Service objects separated by a colon (:).
-pwddn	List of Password service objects serving the realm. The list contains the DNs of the Password service objects separated by a colon (:).
-enctypes	Encryption types supported by the realm. This is a colon-separated list.
-defenctype	Default encryption type for the realm. This is also a part of supported enctypes list.
-salttypes	Salt types supported by the realm. This is a colon-separated list.
-defsalttype	Default salt types for the realm.
-policy	Reference to a policy object (dn) that is applicable to all the principals in a realm.
-up	Specifies to use the universal password of the user as the Kerberos password for the principals in the realm.
-k	Specifies the encryption type of the master key in the database. The default is the type given in the krb5.conf file.
-m	Specifies that the master password should be read from the keyboard rather than from a file or disk.

Parameter	Description
-P	Master password.
-f	Stash file of the master password.
-r	Specifies the Kerberos realm. By default, the default_realm parameter of the krb5.conf file is used.

iManager

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click Kerberos Management > New Realm.

Refer to the iManager online help for more information.

Modifying a Realm

You can modify the realm using either of the following methods:

- ♦ [Command Line \(page 32\)](#)
- ♦ [iManager \(page 34\)](#)

Command Line

Use the following syntax to modify a realm:

```
kdb5_util [-D user_dn] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]

modify [-subtree subtree_dn] [-sscope search_scope]
       [-ldapdn ldap_server_list | [-clearldapdn ldap_server_list]
       [-addldapdn ldap_server_list]] [-kdcn kdc_service_list |
       [-clearkdcn kdc_service_list] [-addkdcn kdc_service_list]]
       [-admindn admin_service_list | [-clearadmindn admin_service_list]
       [-addadmindn admin_service_list]] [-pwddn passwd_service_list |
       [-clearpwddn passwd_service_list] [-addpwddn passwd_service_list]]
       [-enctypes supported_enc_types | [-clearenctypes enc_type_list]
       [-addenctypes enc_type_list]] [-defenctype default_enc_type]
       [-salttypes supported_salt_types | [-clearsalttypes salt_type_list]
       [-addsalttypes salt_type_list]] [-defsalttype default_salt_type]
       [-policy policy_dn|-clearpolicy] [-up|-clearup] [-r realm]
```

For example:

```
kdb5_util -r ATHENA.MIT.EDU -D cn=admin,o=org modify -sscope 1 -clearkdcn
cn=service-kdc1,o=org:cn=service-kdc2,o=org -addkdcn cn=service-
kdc3,o=org:cn=service-kdc4,o=org -enctypes des3-hmac-sha1:des-cbc-md5 -
defenctype des3-hmac-sha1 -addsalttypes v4:special -clearpolicy -up
```

Output of the above command:


```
Password for "cn=admin,o=org":
```


Table 19 Modify Realm Parameter Description

Parameter	Description
-subtree	Subtree containing principals and other Kerberos objects in the realm.
-sscope	Scope for searching the principals under the specified subtree. The parameter sscope specifies the search scope for searching the principals under the subtree specified. The possible values are 1 or one (one level), 2 or sub (subtree).
-ldapdn	List of LDAP servers that the Kerberos servers (KDC and administration servers) can contact. The list contains the DNs of the LDAP servers separated by a colon (:).
-clearldapdn	List of LDAP servers that need to be removed from the list. The list contains the DNs of the LDAP servers separated by a colon (:).
-addldapdn	List of LDAP servers that need to be added to the list. The list contains the DNs of the LDAP servers separated by a colon (:).
-kdc dn	List of KDC service objects serving the realm. The list contains the DNs of the KDC Service objects separated by a colon (:).
-clearkdc dn	List of KDC service objects that need to be removed from the list. The list contains the DNs of the KDC service objects separated by a colon (:).
-addkdc dn	List of KDC service objects that need to be added to the list. The list contains the DNs of the KDC service objects separated by a colon (:).
-admin dn	List of Administration service objects serving the realm. The list contains the DNs of the Administration service objects separated by a colon (:).
-clearadmin dn	List of Administration service objects that need to be removed from the list. The list contains the DNs of the Administration service objects separated by a colon (:).
-addadmin dn	List of Administration service objects that need to be added to the list. The list contains the DNs of the Administration service objects separated by a colon (:).
-pwd dn	List of Password service objects serving the realm. The list contains the DNs of the Password service objects separated by a colon (:).
-clearpwd dn	List of Password service objects that need to be removed from the list. The list contains the DNs of the Administration service objects separated by a colon (:).
-addpwd dn	List of Password service objects that need to be added to the list. The list contains the DNs of the Password service objects separated by a colon (:).
-enctypes	Encryption types supported by the realm. This is a colon-separated list.
-clearenctypes	Encryption types that need to be removed from the supported encryption types list of the realm. This is a a colon-separated list.
-addenctypes	Encryption types that need to be added to the supported encryption types list of the realm. This is a colon-separated list.
-defenctype	Default encryption type for the realm.
-salttypes	Salt types supported by the realm. This is a colon-separated list.
-clearsalttypes	Salt types that need to be removed from the supported salt types list of the realm. This is a colon-separated list.

Parameter	Description
-addsalttypes	Salt types that need to be added to the supported salt types list of the realm. This is a colon-separated list.
-defsalttype	Default salt types for the realm.
-policy	Reference to a policy object (dn) that is applicable to all the principals in a realm.
-up	Specifies to use the universal password of the user as the Kerberos password for the principals in the realm.
-clearup	This attribute specifies not to use the universal password of the user as the Kerberos password.
-r	Specifies the Kerberos realm. By default, the default_realm parameter of the krb5.conf file is used.

iManager

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click Kerberos Management > Edit Realm.

Refer to the iManager online help for more information.

Modifying the subtree for a realm

If you change the subtree for a realm, then the Kerberos service objects (KDC, Admin Server and Password Server) are not automatically re-assigned with the appropriate rights.

For example,

If your subtree for the realm is "ou=students,o=mit" for the realm "ATHENA.MIT.EDU" and you change it to "o=mit", then the service objects that represent the Kerberos services for your realm are not automatically assigned rights over the new subtree.

To reassign the rights over the new subtree, follow these steps:

- 1 Stop the Kerberos services.
- 2 Destroy the service objects.
- 3 Create the service objects again.
- 4 Start the Kerberos services

Modifying the search scope for a realm

If you modify the sscope for a realm, then the objects created previously under the old scope will still exist.

For example,

If your subtree is "o=mit" that has a container "ou=students,o=mit" and you change the search scope from "sub" to "one", the Kerberos principal objects that were created under "ou=students,o=mit" will still exist.

Viewing a Realm

Use the following syntax to view realms:

```
kdb5_util [-D user_dn] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]

view      [-r realm]
```

For more information on the parameters, refer to [Table 18, “Create Realm Parameter Description,” on page 31](#).

For example:

```
kdb5_util -D cn=admin,o=org -h ldap-server1.mit.edu view -r ATHENA.MIT.EDU
```

Output of the above command:

```
Password for "cn=admin,o=org":
    Realm Name: ATHENA.MIT.EDU
    Subtree: ou=hr,o=org
    SearchScope: SUB
    KDC Services: cn=service-kdc1,o=org
    Admin Services: cn=admin-service,o=org
    Supported Enc Types: DES cbc mode with CRC-32
                        DES cbc mode with RSA-MD4
                        DES cbc mode with RSA-MD5
                        Triple DES cbc mode with HMAC/sha1
                        AES-128 CTS mode with 96-bit SHA-1 HMAC
                        AES-256 CTS mode with 96-bit SHA-1 HMAC
                        ArcFour with HMAC/md5
    Default Enc Type: Triple DES cbc mode with HMAC/sha1
    Supported Salt Types: Version 5
                        Version 4
                        Version 5 - No Realm
                        Version 5 - Realm Only
                        Special
    Default Salt Type: Version 5
```

Destroying a Realm

You can destroy a realm using either of the following methods:

- ◆ [Command Line \(page 35\)](#)
- ◆ [iManager \(page 36\)](#)

Command Line

Use the following syntax to destroy a realm:

```
kdb5_util [-D user_dn] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]

destroy [-f] [-r realm]
```

For more information on the parameters, refer to [Table 18, “Create Realm Parameter Description,” on page 31](#).

For example:


```
kdb5_util -D cn=admin,o=org -h ldap-server1.mit.edu destroy -r ATHENA.MIT.EDU
```

Output of the above command:

```
Password for "cn=admin,o=org":
Deleting KDC database of 'ATHENA.MIT.EDU', are you sure?
(type 'yes' to confirm)? yes
OK, deleting database of 'ATHENA.MIT.EDU'...
** Database of 'ATHENA.MIT.EDU' destroyed.
```

The principals associated with this realm are also deleted.

iManager

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click Kerberos Management > Delete Realm.

Refer to the iManager online help for more information.

Listing Realms

Use the following syntax to list realms:

```
kdb5_util [-D user_dn] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]
```

```
list
```

For more information on the parameters, refer to [Table 18, “Create Realm Parameter Description,” on page 31](#).

For example:

```
kdb5_util -D cn=admin,o=org list
```

Output of the above command:

```
Password for "cn=admin,o=org":
NovellRealm
MYREALM
ATHENA.MIT.EDU
MEDIA-LAB.MIT.EDU
```

Managing Services

You can manage the KDC, Administration, and Password services using the `kdb5_util` command. This section provides information about the following:

- ◆ [“Creating a Service” on page 37](#)
- ◆ [“Modifying a Service” on page 38](#)
- ◆ [“Viewing a Service” on page 39](#)
- ◆ [“Listing Services” on page 39](#)
- ◆ [“Destroying a Service” on page 40](#)
- ◆ [“Setting a Password for Service Objects” on page 41](#)
- ◆ [“Setting the Server Certificate” on page 42](#)

Creating a Service

You can create a service using either of the following methods:

- ◆ [Command Line \(page 37\)](#)
- ◆ [iManager \(page 38\)](#)

Command Line

Use the following syntax to create a service using `kdb5_util`:

```
kdb5_util [-D user_dn] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]

create_service {-kdc|-admin|-pwd} [-servicehost service_host_list]
              [-realm realm_list]
              [-randpw|-fileonly] [-f filename] service_dn
```

The service is created or modified in eDirectory™.

For example:

```
kdb5_util -D cn=admin,o=org create_service -kdc -randpw -f /home/andrew/
conf_keyfile cn=service-kdc,o=org
```

Output of the above command is similar to the following:


```
Password for "cn=admin,o=org":
File does not exist. Creating the file /home/andrew/conf_keyfile...
```

The following table describes the configuration parameters of `create_service` option of the `kdb5_util` command:

Table 20 **create_service Parameter Description**

Parameter	Description
-kdc	KDC service
-admin	Administration service
-pwd	Password service
-servicehost	List of entries separated by a colon (:) where each entry consists of the hostname or IP address of the server hosting the service, transport protocol, and the port number of the service separated by a pound sign (#). For example, server1#tcp#88:server2#udp#89.
-realm	List of realms that can be serviced by Kerberos. The list contains the names of the realms separated by a colon (:).
-randpw	Generate and set a random password. This option cannot be specified with -fileonly option. This option will not work when Universal Password is enabled.
-fileonly	Set the password in the service password file only, without updating the directory object. This is useful when the service object is shared by multiple hosts.
-f	Complete path of the service password file where the Service object password is stashed. The default path is /usr/local/var/service/passwd.
servicedn	dn of the Kerberos service to be created.

iManager

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Select Kerberos Management > New Service.

Refer to the iManager online help for more information.

Modifying a Service

You can modify a service using either of the following methods:

- ♦ [Command Line \(page 38\)](#)
- ♦ [iManager \(page 39\)](#)

Command Line

Use the following syntax to modify a service:

```
kdb5_util [-D user_dn [-w passwd]] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]

modify_service [-servicehost service_host_list |
              [-clearservicehost service_host_list]
              [-addservicehost service_host_list]]
              [-realm realm_list | [-clearrealm realm_list]
              [-addrealm realm_list]] service_dn
```

This command modifies the attributes of a service and assigns appropriate rights.

For example:

```
kdb5_util -D cn=admin,o=org -w passwd modify_service -realm ATHENA.MIT.EDU
cn=service-kdc,o=org
```

Output of the above command will be similar to the following:

```
Password for "cn=admin,o=org":
Changing rights for the service object. Please wait ... done
```


The following table describes the modify_service parameters:

Table 21 modify_service Parameter Options

Parameter	Description
-servicehost	List of entries separated by a colon (:) where each entry consists of host name or IP Address of the Server hosting the Service, transport protocol, and port number of the Service separated by a pound sign (#). For example, server1#tcp#88:server2#udp#89.
-clearservicehost	List of servicehost entries to be removed from the existing list. This is a colon-separated where each entry consists of host name or IP Address of the Server hosting Service, transport protocol, and port number of the Service separated by a pound sign (#).
-addservicehost	List of servicehost entries to be added to the existing list. This is a colon-separated list where each entry consists of host name or IP Address of the Server hosting Service, transport protocol, and port number of the service separated by a pound sign (#).

Parameter	Description
-realm	List of realms that are associated with this service. The list contains the names of the realms separated by a colon (:).
-clearrealm	List of realms to be removed from the existing list. The list contains the names of the realms separated by a colon (:).
-addrealm	List of realms to be added to the existing list. The list contains the names of the realms separated by a colon (:).
servicedn	DN of the Kerberos service to be modified.

iManager

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Select Kerberos Management > Edit Service.

Refer to the iManager online help for more information.

Viewing a Service

Use the following syntax to view a service:

```
kdb5_util [-D user_dn [-w passwd]] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]
```

```
view_service service_dn
```

For example:

```
kdb5_util -D cn=admin,o=org view_service cn=kdc-service1,o=org
```

Output of the above command will be similar to the following:

```
Password for "cn=admin,o=org":
    Service dn: cn=service-kdc,o=org
    Service type: kdc
Service host list:
    Realm DN list:
```

Table 22 view_service Parameter Description

Parameter	Description
servicedn	DN of the Kerberos service to be viewed.

Listing Services

Use the following syntax to list a service:

```
kdb5_util [-D user_dn [-w passwd]] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]
```

```
list_service [-basedn base_dn]
```

For more information on the parameters, refer to [Table 20, “create_service Parameter Description,” on page 37](#).

Table 23 list_service Parameter Description

Parameter	Description
-basedn	Specifies the base DN for searching the services. The basedn option is made available to limit the search to a particular subtree. If this option is not provided, the entire tree will be searched, which means that the default value for the base DN is root. Therefore, this option is suitable in scenarios where the tree is distributed over more than one geographical location.

This command lists the name of all existing services.

For example:

```
kdb5_util -D cn=admin,o=org list_service
```

The output of the above command is similar to the following:

```
Password for "cn=admin,o=org":
cn=service-kdc,o=org
cn=service-adm,o=org
cn=service-pwd,o=org
```

Destroying a Service

You can destroy a service using either of the following methods:

- ◆ [Command Line \(page 40\)](#)
- ◆ [iManager \(page 41\)](#)

Command Line

Use the following syntax to destroy a service:

```
kdb5_util [-D user_dn [-w passwd]] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]

destroy_service [-force] [-f stashfilename] service_dn
```

For more information on the parameters, refer to [Table 20, “create_service Parameter Description,” on page 37](#).

The -f option becomes necessary if you have chosen to use a stash file of your choice while creating the service or setting the password for it. If this option is not provided, the entry for the service to be destroyed will be looked up in the default stash file. Therefore, though the service object gets destroyed, the entry might remain in the stash file of your choice.


For example:

```
kdb5_util -D cn=admin,o=org destroy_service cn=service-kdc,o=org
```

Output of the above command is similar to the following:

```
Password for "cn=admin,o=org":
This will delete the service object 'cn=service-kdc,o=org', are you sure?
(type 'yes' to confirm)? yes
** service object 'cn=service-kdc,o=org' deleted.
```


iManager

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Click Kerberos Management > Delete Service.

Refer to the iManager online help for more information.

Setting a Password for Service Objects

You can set a password for service objects such as KDC, Administration, and Password server in eDirectory and store them in a file. The `-fileonly` option stores the password in a file and not in the eDirectory object.

```
kdb5_util [-D user_dn [-w passwd]] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]
setsrvpw [-randpw|-fileonly] [-f filename] service_dn
```

For example:

```
kdb5_util setsrvpw -fileonly -f /home/andrew/conf_keyfile
cn=service-kdc,o=org
```

If you do not specify a filename, the default path `/usr/local/var/service_passwd` is used.

`kdb5_util` does not store the password in plain text format in the file. It is encrypted using a unique machine-dependant key and then stored in the file.

IMPORTANT: The password file should not be edited manually. It must be modified using `kdb5_util` only. Also, because passwords in this file are encrypted using a unique machine-dependant key, the password file becomes unusable if it is moved to a different machine.

The following table describes the configuration parameters:

Table 24 setsrvpw Parameter Description

Parameter	Description
-D	Distinguished name of the user who has sufficient rights to authenticate to the LDAP server.
-w	Specifies the userdn password. This is not recommended.
-h	Host name or IP Address of the server hosting LDAP service for a Kerberos realm.
-p	SSL port number of the LDAP server.
-t	Specifies the filename that contains Trusted Root Certificate of the LDAP server.
-randpw	Generates and sets a random password. You can specify this option if you want to store the password both in eDirectory and a file. You cannot use the <code>-fileonly</code> option when you specify <code>-randpw</code> .
-fileonly	Stores the password only in a file and not in eDirectory. You cannot use the <code>-randpw</code> option when you specify <code>-fileonly</code> .
-f	Complete path of the service password file.
servicedn	DN of the service object whose password is to be set.

Setting the Server Certificate

This section describes the steps to configure the Kerberos services (KDC, Administration and Password servers) for authenticating to eDirectory using LDAP SASL EXTERNAL (CertMutual) authentication.

To set up certificate-based authentication, complete the following procedure:

- 1** Create a new directory. For example, *kerbcert*.
- 2** Create a file called *openssl.cnf* in the *kerbcert* directory with the following contents:

```
[ req ]
distinguished_name = req_distinguished_name
prompt = no
```

```
[ req_distinguished_name ]
CN=service-kdc.O=org
```

Replace CN=service-kdc.O=org with the FDN of the service object in eDirectory.

NOTE: The attribute names 'CN', 'OU', 'O' must be in upper case. The components of the FDN must be separated by '.'(dot) and not by ','(comma).

- 3** Change directory

```
cd kerbcert/
```

- 4** Create a private key and certificate signing request (CSR).

- 4a** Enter the following command:

```
openssl req -newkey rsa:1024 -keyout key.pem -out req.pem -
config openssl.cnf
```

The private key will be written to *key.pem* and the certificate signing request to *req.pem*. For more information, refer to the [OpenSSL Website \(http://www.openssl.org/docs/apps/openssl.html\)](http://www.openssl.org/docs/apps/openssl.html).

- 4b** Specify the password at the prompt.

This password protects the private key.

- 5** Connect to the eDirectory tree using iManager and issue a certificate as described in the [Novell Certificate Server 2.21 Administration Guide \(http://www.novell.com/documentation/crt221ad/index.html\)](http://www.novell.com/documentation/crt221ad/index.html).

When prompted for the certificate signing request, specify the *req.pem* file path.

Export the issued certificate in base 64 format (.b64) into a file called *cert.b64* in the new directory (*kerbcert* in our example).

- 6** Concatenate the files *key.pem* and *cert.b64* into a single file *cert-key.pem* as follows:

```
cat key.pem cert.b64 > cert-key.pem
```

- 7** Configure the service to use the issued certificate for authentication instead of the password as follows:

```
kdb5_util setsrvcert -f <path_of_the_password_stash_file>
-cert cert-key.pem <service_dn>
```

service_dn should be the FDN specified in the *openssl.cnf* file (CN=service-kdc.O=org as per our example).

Enter the password, when you are prompted to do so. This password is same as the one you had given in [Step 4b](#).

The service is now configured to use certificate-based authentication instead of password-based authentication.

Before starting the service, configure eDirectory to accept certificate-based authentication as follows:

- 1 Modify the LDAP server SSL/TLS configuration using iManager or ConsoleOne as follows:

Change the Client Certificate field from Not requested to Requested as described in [Novell eDirectory 8.7.3 Administration Guide](#) (<http://www.novell.com/documentation/edir873/edir873/data/agtzhz5.html#agtzhz5>).

- 2 Check whether the SASL EXTERNAL mechanism is installed as follows:

```
ldapsearch -x -h <eDirectory_host_name> -b "" -s base | grep
'supportedSASLMechanisms'
```

The SASL mechanisms supported by eDirectory will be listed. Check if the EXTERNAL mechanism is in the list. If not, the mechanism has to be installed as described in [Novell eDirectory 8.7.3 Administration Guide](#) (<http://www.novell.com/documentation/edir873/edir873/data/agtzhz5.html#agtzhz5>).

Managing Principals

You can manage principals through kadmin. This section explains the following:

- ♦ “Adding a Principal” on page 43
- ♦ “Modifying a Principal” on page 47
- ♦ “Deleting a Principal” on page 47
- ♦ “Listing Principals” on page 48
- ♦ “Getting Principal Information” on page 48
- ♦ “Setting Principal Password” on page 49
- ♦ “Extracting Principal Key to a Keytab File” on page 50
- ♦ “Removing Keytab Entry” on page 50

Adding a Principal

User and the service principals can be created only within the realm subtree and its sub-containers. However, the service principals can even be created within the realm container by specifying the container DN option with the realm container while creation of the service principal.

You can add a principal using either of the following methods:

- ♦ “Command Line” on page 43
- ♦ “iManager” on page 46

Command Line

To create a principal, enter the following at the kadmin prompt:

```

add_principal [options] principal
options are:
    [-x db_princ_args] [-expire expdate] [-pwexpire pwexpdate] [-maxlife
maxtixlife]
    [-kvno kvno] [-policy policy] [-randkey] [-pw password]
    [-maxrenewlife maxrenewlife]
    [-e keysaltlist]
    [(+|-)attribute]
attributes are:
    allow_postdated allow_forwardable allow_tgs_req allow_renewable
    allow_proxiable allow_dup_skey allow_tix requires_preauth
    requires_hwauth needchange allow_svr password_changing_service

```

Table 25 add_principal Parameter Description

Parameter	Description
-x	Denotes the database-specific options. The following are the options for LDAP as the backend: <ul style="list-style-type: none"> ◆ -x userdn=<userdn> Specifies the associated eDirectory user object while creating a Kerberos user principal. ◆ -x up=<on off clr> Specifies if the Kerberos User Principal associated with the eDirectory user object will make use of the universal password. ◆ -x tktpolicydn Associates a ticket policy object to the Kerberos principal. ◆ -x containerdn=<container_dn> Specifies the eDirectory container under which the Kerberos service principal is to be created.
-expire	Specifies the expiration date of the principal
-pwexpire	Specifies the password expiration date
-maxlife	Specifies the maximum ticket life for the principal
-kvno	Explicitly sets the key version number.
-policy	Specifies the password policy used by this principal. If no policy is supplied, then if the policy "default" exists and the -clearpolicy is also not specified, then the policy "default" is used; otherwise, the principal will have no password policy, and a warning message will be printed.
-randkey	Sets the key of the principal to a random value. Do not use this while creating InterRealm principals.
-pw	Sets the key of the principal to the specified string and does not prompt for a password. WARNING: Using this option at the shell prompt can be risky if unauthorized users gain read access to the script.
-maxrenewlife	Specifies the maximum renewable life of tickets for the principal.

Parameter	Description
-e	Uses the specified list of enctype-salttype pairs for setting the key of the principal. The quotes are necessary if there are multiple enctype-salttype pairs. This will not function against kadmin daemons earlier than krb5-1.2. NOTE: If universal password integration is enabled, refer to “Key Generation” on page 63 .
-clearpolicy	Prevents the policy "default" from being assigned when (-) policy is not specified. This option has no effect if the policy "default" does not exist.
{- +}allow_postdated	(-) allow_postdated prohibits this principal from obtaining postdated tickets. (Sets the KRB5_KDB_DISALLOW_POSTDATED flag.) (+) allow_postdated clears this flag.
{- +}allow_forwardable	(-) allow_forwardable prohibits this principal from obtaining forwardable tickets. (Sets the KRB5_KDB_DISALLOW_FORWARDABLE flag.) (+) allow_forwardable clears this flag.
{- +}allow_renewable	(-) allow_renewable prohibits this principal from obtaining renewable tickets. (Sets the KRB5_KDB_DISALLOW_RENEWABLE flag.) (+) allow_renewable clears this flag.
{- +}allow_proxiable	(-) allow_proxiable prohibits this principal from obtaining proxiable tickets. (Sets the KRB5_KDB_DISALLOW_PROXIABLE flag.) (+) allow_proxiable clears this flag.
{- +}allow_dup_key	(-) allow_dup_key disables user-to-user authentication for this principal by prohibiting this principal from obtaining a session key for another user. (Sets the KRB5_KDB_DISALLOW_DUP_SKEY flag.) (+) allow_dup_key clears this flag.
{- +}requires_preauth	(+) requires_preauth requires this principal to preauthenticate before being allowed to kinit. (Sets the KRB5_KDB_REQUIRES_PRE_AUTH flag.) (-) requires_preauth clears this flag.
{- +}requires_hwauth	(+) requires_hwauth requires this principal to preauthenticate using a hardware device before being allowed to kinit. (Sets the KRB5_KDB_REQUIRES_HW_AUTH flag.) (-) requires_hwauth clears this flag.
{- +}allow_svr	(-) allow_svr prohibits the issuance of service tickets for this principal. (Sets the KRB5_KDB_DISALLOW_SVR flag.) (+) allow_svr clears this flag.
{- +}allow_tgs_req	(-) allow_tgs_req specifies that a Ticket-Granting Service (TGS) request for a service ticket for this principal is not permitted. (+) allow_tgs_req clears this flag. The default is (+) allow_tgs_req. In effect, (-) allow_tgs_req sets the KRB5_KDB_DISALLOW_TGT_BASED flag on the principal in the database.
{- +}allow_tix	(-) allow_tix forbids the issuance of any tickets for this principal. (+) allow_tix clears this flag. The default is (+) allow_tix. In effect, (-) allow_tix sets the KRB5_KDB_DISALLOW_ALL_TIX flag on the principal in the database.
{- +}needchange	(+) needchange sets a flag in attributes field to force a password change; (-) needchange clears it. The default is (-) needchange. In effect, (+) needchange sets the KRB5_KDB_REQUIRES_PWCHANGE flag on the principal in the database.

Parameter	Description
{- +}password_changing_service	(+) password_changing_service sets a flag in the attributes field marking this as a password change service principal. (-) password_changing_service clears the flag. This flag intentionally has a long name. The default is (-) password_changing_service. In effect, (+) password_changing_service sets the KDB_PWCHANGE_SERVICE flag on the principal in the database.

Creating User Principal

Every Kerberos user principal is associated with the eDirectory object. Therefore, while creating a Kerberos user principal, the associated eDirectory user object must be mentioned.

To create a user principal, enter the following at the kadmin prompt:

```
add_principal -x up=on -x userdn=cn=user1,o=org user_princ
```

If the userdn is not present in eDirectory, it creates a new one with the specified name.

The output of the above command is similar to the following:

```
WARNING: no policy specified for user_princ@MYREALM; defaulting to no policy
Enter password for principal "user_princ@MYREALM":
Re-enter password for principal "user_princ@MYREALM":
Principal "user_princ@MYREALM" created.
```

Creating a Service Principal


To create a service principal, enter the following:

```
add_principal -x containerdn=ou=sales,o=org service_princ
```

The output of the above command is similar to the following:

```
WARNING: no policy specified for service_princ@MYREALM; defaulting to no
policy
Enter password for principal "service_princ@MYREALM":
Re-enter password for principal "service_princ@MYREALM":
Principal "service_princ@MYREALM" created.
```

iManager

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Select Kerberos Management > New Principal.

Refer to the iManager online help for more information.

Associating a Ticket Policy to the Kerberos Principal

A ticket policy object can be associated with a Kerberos principal using the add_principal command of the kadmin utility.

For example:

```
add_principal -x tktpolicydn=cn=tktpolicy,o=org serviceuser
```

Modifying a Principal

You can modify a principal using either of the following methods:

- ◆ [Command Line \(page 47\)](#)
- ◆ [iManager \(page 47\)](#)

Command Line

To modify principals, enter the following at the kadmin command prompt:

```
modify_principal [options] principal
```

options are:

```
[-x db_princ_args]* [-expire expdate] [-pwexpire pwexpdate] [-maxlife
maxtixlife]
[-kvno kvno] [-policy policy] [-clearpolicy]
[-maxrenewlife maxrenewlife] [(+|-)attribute]
```

attributes are:

```
allow_postdated allow_forwardable allow_tgs_req allow_renewable
allow_proxiabile allow_dup_skey allow_tix requires_preauth
requires_hwauth needchange allow_svr password_changing_service
```

For details about the parameters, refer to [Table 25, “add_principal Parameter Description,” on page 44](#).


For example:

```
modify_principal -x up=off -policy cn=realm_policy,o=org +requires_preauth
princ
```

The output of the above command is similar to the following:

```
Principal "princ@MYREALM" modified.
```

iManager

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Select Kerberos Management > Edit Principal.

Refer to the iManager online help for more information.

Associating a Ticket Policy to the Kerberos Principal

If the principal is already created, use the `modify_principal` command of `kadmin` utility.

For example:

```
modify_principal -x tktpolicydn=cn=tktpolicy,o=org serviceuser
```

Deleting a Principal

You can delete a principal using either of the following methods:

- ◆ [Command Line \(page 48\)](#)
- ◆ [iManager \(page 48\)](#)

Command Line

To delete a principal, enter the following at the kadmin command prompt:

```
delete_principal [-force] principal
```

If the `-force` option is not specified, you are prompted to confirm the deletion. The `delete_principal` command will not delete the user but only the Kerberos attribute.


For example:

```
delete_principal princ1
```

The output of the above command is similar to the following:

```
Are you sure you want to delete the principal "princ1@MYREALM"? (yes/no): yes
Principal "princ1@MYREALM" deleted.
Make sure that you have removed this principal from all ACLs before reusing.
```

iManager

1 In Novell iManager, click the Roles and Tasks button .

2 Select Kerberos Management > Delete Principal.

Refer to the iManager online help for more information.

Listing Principals

To list principals, enter the following at the kadmin prompt:

```
list_principals [expression]
```

For example:

```
list_principals princ*
```

The output of the above command is similar to the following:

```
princ@MYREALM
princ1@MYREALM
princ2@MYREALM
```

Getting Principal Information

To get the attributes of a principal, enter the following at the kadmin command prompt:

```
get_principal [-terse] principal
```

For example:

```
get_principal user_princ
```

The output of the above command is similar to the following:

```
Principal: user_princ@MYREALM
Expiration date: [never]
Last password change: Tue May 31 13:55:24 IST 2005
Password expiration date: [none]
Maximum ticket life: 1 day 00:00:00
Maximum renewable life: 7 days 00:00:00
Last modified: Tue May 31 14:05:06 IST 2005 (CN=service-adm,O=org@MYREALM)
```



```

Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 1
Key: vno 1, Triple DES cbc mode with HMAC/sha1, no salt
Attributes: REQUIRES_PRE_AUTH
Policy: [none]

```

Setting Principal Password

You can set principal password using either of the following methods:

- ◆ [Command Line \(page 49\)](#)
- ◆ [iManager \(page 49\)](#)

Command Line

To change the password of a principal, enter the following at the kadmin prompt:

```
change_password [-randkey] [-keepold] [-e keysaltlist] [-pw password]
principal
```

Table 26 **change_password Parameter Description**

Parameter	Description
-randkey	Sets the key of the principal to a random value.
-keepold	Keeps the previous kvno's keys. There is no easy way to delete the old keys, and this flag is usually not necessary except perhaps for TGS keys. Don't use this flag unless you are sure you want to use it.
-e	Uses the specified list of enctype-salttype pairs for setting the key of the principal. The quotes are necessary if there are multiple enctype-salttype pairs. NOTE: If universal password integration is enabled, refer to "Key Generation" on page 63 .
-pw	Sets the password to the specified string. We do not recommend you to use it.

For example:

```
change_password princ2
```

The output of the above command is similar to the following:

```

Enter password for principal "princ2":
Re-enter password for principal "princ2":
Password for "princ2@MYREALM" changed.


```

```
change_password -pw secret princ2
```

The output of the above command is similar to the following:

```
Password for "princ2@MYREALM" changed.
```

iManager

- 1 In Novell iManager, click the Roles and Tasks button 

2 Select Kerberos Management > Set Principal Password.

Refer to the iManager online help for more information.

Extracting Principal Key to a Keytab File

To extract the principal key to a keytab file, enter the following command at the kadmin prompt:

```
ktadd [-keytab keytab] [-q] [-e keysaltlist] [principal | -glob princ-exp]
[...]
```

Table 27 ktadd Parameter Description

Parameter	Description
-keytab	Specifies the keytab file path.
-q	Displays less verbose status information.
-e	Uses the specified list of enctype-salttype pairs for setting the key of the principal. The quotes are necessary if there are multiple enctype-salttype pairs.

NOTE: If universal password integration is enabled, refer to ["Key Generation" on page 63](#).

For example:

```
ktadd -k /etc/key-tab user_princ
```

The output of the above command is similar to the following:

```
Entry for principal user_princ with kvno 2, encryption type Triple DES cbc
mode with HMAC/sha1 added to keytab WRFILE:/etc/key-tab.
```

Removing Keytab Entry

To remove entries from a keytab, enter the following command at the kadmin prompt:

```
ktremove [-keytab keytab] [-q] principal [kvno|"all"|"old"]
```

Table 28 kremove Parameter Description

Parameter	Description
-keytab	Specifies the keytab file path.
-q	Displays less verbose status information.

For example:

```
ktremove -k /etc/key-tab user_princ all
```

The output of the above command is similar to the following:

```
Entry for principal user_princ with kvno 2 removed from keytab WRFILE:/etc/
key-tab.
```

Managing Ticket Policies

The policy objects stored in eDirectory can be attached to Kerberos principals, realms, or even the Kerberos container. Policy-related attributes can also be associated directly with the user or realm but are not explained here.

- ◆ [“Creating a Ticket Policy” on page 51](#)
- ◆ [“Modifying a Ticket Policy” on page 53](#)
- ◆ [“Destroying a Ticket Policy” on page 53](#)
- ◆ [“Viewing a Ticket Policy” on page 54](#)
- ◆ [“Listing Ticket Policies” on page 55](#)

Creating a Ticket Policy

You can add a Ticket Policy using either of the following methods:

- ◆ [Command Line \(page 51\)](#)
- ◆ [iManager \(page 52\)](#)

Command Line

Use the following command to add a ticket policy:

```
kdb5_util [-D user_dn [-w passwd]] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]

create_policy [-maxtktlife max_ticket_life]
              [-maxrenewlife max_renewable_ticket_life] [ticket_flags] policy_dn
```

For example:

```
kdb5_util -D cn=admin,o=org -h ldap-server1.mit.edu -p 636 create_policy
-maxtktlife "1 day" -maxrenewlife "1 week"
-allow_postdated +needchange -allow_forwardable cn=tktpolicy,o=org
```


Refer to the following table for the description of the parameters:

Table 29 create_policy Parameter Description

Parameter	Description
-D	Distinguished name of the user who has sufficient rights to authenticate to the LDAP server.
-w	Specifies the userdn password. This is not recommended.
-h	Host name or IP Address of the server hosting LDAP service for a Kerberos realm.
-p	SSL port number of the LDAP server.
-t	Specifies the filename that contains Trusted Root Certificate of the LDAP server.
-maxtklife	Specifies the maximum life-time of ticket issued.
-maxrenewlife	Specifies the maximum life-time to which issued ticket can be renewed.

Parameter	Description
ticket_flags	<p>Specifies the ticket flags. If this option is not specified, by default, none of the flags are set. This means that all the ticket options will be allowed and no restriction will be set.</p> <p>The various flags are:</p>
{- +}allow_postdated	allows (+) principals to obtain postdated tickets / prohibits (-) principals from obtaining postdated tickets.
{- +}allow_forwardable	allows (+) principals to obtain forwardable tickets / prohibits (-) principals from obtaining forwardable tickets.
{- +}allow_renewable	allows (+) principals to obtain renewable tickets / prohibits (-) principals from obtaining renewable tickets.
{- +}allow_proxiable	allows (+) principals to obtain proxiable tickets / prohibits (-) principals from obtaining proxiable tickets.
{- +}allow_dup_skey	disables (-) / enables (+) user-to-user authentication for principals, by respectively prohibiting / allowing obtaining of a session key for another user.
{- +}requires_preauth	makes principals require (+) / not require (-) pre-authentication before being allowed to 'kinit'.
{- +}requires_hwauth	makes principals require (+) / not require (-) pre-authentication using a hardware device before being allowed to 'kinit'.
{- +}allow_svr	allows (+) / prohibits (-) issuance of service tickets for this principal.
{- +}allow_tgs_req	(-)allow_tgs_req specifies that a Ticket-Granting Service (TGS) request for a service ticket for this principal is not permitted. (+) allow_tgs_req clears this flag. The default is (+) allow_tgs_req. In effect, (-)allow_tgs_req sets the KRB5_KDB_DISALLOW_TGT_BASED flag on the principal in the database.
{- +}allow_tix	allows (+) / prohibits (-) issuance of any tickets for principals. The default is "+allow_tix".
{- +}needchange	makes principals require (+) / not require (-) a password change.
{- +}password_changing_service	used to set (+) / unset(-) principals as password changing services.
policy_dn	Distinguished name of the policy.

iManager

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Select Kerberos Management > New Policy.

Refer to the iManager online help for more information.

Modifying a Ticket Policy

You can modify a ticket policy using either of the following methods:

- ♦ “Command Line” on page 53
- ♦ “iManager” on page 53

Command Line

Use the following command to modify a ticket policy:

```
kdb5_util [-D user_dn [-w passwd]] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]
```


```
modify_policy [-maxttlifetime max_ticket_life]
              [-maxrenewlife max_renewable_ticket_life] [ticket_flags] policy_dn
```

For more information on the parameters, refer to [Table 29, “create_policy Parameter Description,” on page 51](#).

For example:

```
kdb5_util -D cn=admin,o=org -h ldap-server1.mit.edu -p 636 modify_policy -
maxttlifetime "60 minutes" -maxrenewlife "10 hours" +allow_postdated -
requires_preauth cn=tktpolicy,o=org
```

iManager

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Select Kerberos Management > Edit Policy.

Refer to the iManager online help for more information.

Destroying a Ticket Policy

You can destroy a ticket policy using either of the following methods:

- ♦ “Command Line” on page 53
- ♦ “iManager” on page 54

Command Line

Use the following command to destroy a ticket policy:

```
kdb5_util [-D user_dn [-w passwd]] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]
```

```
destroy_policy [-force] policy_dn
```


For example:

```
kdb5_util -D cn=admin,o=org -h ldap-server1.mit.edu -p 636 destroy_policy
-force cn=tktpolicy,o=org
```

Table 30 **destroy_policy Parameter Description**

Parameter	Description
-D	Distinguished name of the user who has sufficient rights to authenticate to the LDAP server.
-w	Specifies the userdn password. This is not recommended.
-h	Host name or IP Address of the server hosting LDAP service for a Kerberos realm.
-p	SSL port number of the LDAP server.
-t	Specifies the filename that contains Trusted Root Certificate of the LDAP server.
-force	Forces the deletion of the policy object. If you do not specify this option, you will be prompted for confirmation while deleting the policy. Enter YES to confirm the deletion.
policy_dn	Distinguished name of the policy.

iManager

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Select Kerberos Management > Delete Policy.

Refer to the iManager online help for more information.

Viewing a Ticket Policy

Use the following command to view a ticket policy:

```
kdb5_util [-D user_dn [-w passwd]] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]
```

```
view_policy policy_dn
```

For example:

```
kdb5_util -D cn=admin,o=org -h ldap-server1.mit.edu -p 636 view_policy
cn=tktpolicy,o=org
```

The expected output will be:

```
Policy: tktpolicy
Maximum ticket life: 0 days 00:60:00
Maximum renewable life: 0 days 10:00:00
Ticket flags: DISALLOW_FORWARDABLE REQUIRES_PWCHANGE
```

Table 31 **view_policy Parameter Description**

Parameter	Description
-D	Distinguished name of the user who has sufficient rights to authenticate to the LDAP server.
-w	Specifies the userdn password. This is not recommended.
-h	Host name or IP Address of the server hosting LDAP service for a Kerberos realm.

Parameter	Description
-p	SSL port number of the LDAP server.
-t	Specifies the filename that contains Trusted Root Certificate of the LDAP server.
policy_dn	Distinguished name of the policy.

Listing Ticket Policies

Use the following command to list policies:

```
kdb5_util [-D user_dn [-w passwd]] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]
```

```
list_policy [-basedn base_dn]
```

For example:

```
kdb5_util -D cn=admin,o=org -h ldap-server1.mit.edu -p 636 list_policy
-basedn o=org
```

The expected output will be as follows:

```
cn=ktktpolicy,o=org
cn=ktktpolicy2,o=org
cn=ktktpolicy3,o=org
```

Table 32 list_policy Parameter Description

Parameter	Description
-D	Distinguished name of the user who has sufficient rights to authenticate to the LDAP server.
-w	Specifies the userdn password. This is not recommended.
-h	Host name or IP Address of the server hosting LDAP service for a Kerberos realm.
-p	SSL port number of the LDAP server.
-t	Specifies the filename that contains Trusted Root Certificate of the LDAP server.
-basedn	Specifies the base DN for searching the policies. The 'basedn' option is made available to limit the search to a particular subtree. If this option is not provided, the entire tree will be searched, which means that the default value for the base DN is root. Therefore, this option is suitable in scenarios where the tree is distributed over more than one geographical location.

Managing Password Policies

The policy management commands in MIT kadmin utility were modified to work with an LDAP directory. The policies control the password of the Kerberos principals. The Kerberos password policies come into effect only when the Kerberos passwords of the principals are different from

the eDirectory user passwords. When the Kerberos passwords are the same as the user's passwords, NSPM password policy is effective.

- ◆ [“Adding a Password Policy” on page 56](#)
- ◆ [“Modifying a Password Policy” on page 56](#)
- ◆ [“Deleting a Password Policy” on page 57](#)
- ◆ [“Viewing Policy Values” on page 58](#)
- ◆ [“Listing Policies” on page 58](#)

Adding a Password Policy

You can add a password policy using either of the following methods:

- ◆ [Command Line \(page 56\)](#)
- ◆ [iManager \(page 56\)](#)

Command Line

This command creates a password policy object, with the 'policy' argument referring to the DN of the same:

```
add_policy [-maxlife time] [-minlife time] [-minlength length] [-minclasses number] [-history number] policy
```


Table 33 add_policy Parameter Description

Parameter	Description
-maxlife	Sets the maximum lifetime of a password
-minlife	Sets the minimum lifetime of a password
-minlength	Sets the minimum length of a password
-minclasses	Sets the minimum number of character classes allowed in a password
-history	Sets the number of past keys kept for a principal

For example, enter the following at the kadmin prompt:

```
add_policy -maxlife "2 days" -minlength 5 cn=realm-policy,o=org
```

iManager

- 1** In Novell iManager, click the Roles and Tasks button .
- 2** Select Kerberos Management > New Password Policy.

Refer to the iManager online help for more information.

Modifying a Password Policy

You can modify the password policy using either of the following methods:

- ◆ [Command Line \(page 57\)](#)

- ♦ [iManager \(page 57\)](#)

Command Line

To modify a policy, enter the following at the kadmin prompt:


```
modify_policy [-maxlife time] [-minlife time] [-minlength length] [-minclasses number] [-history number] policy
```

For more information on the options, refer to [Table 33, “add_policy Parameter Description,” on page 56](#).

For example, enter the following at the kadmin prompt:

```
modify_policy -minlife "1 day" -minclasses 2 cn=realm-policy,o=org
```

iManager

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Select Kerberos Management > Edit Password Policy.

Refer to the iManager online help for more information.

Deleting a Password Policy

You can delete a password policy using either of the following methods:

- ♦ [Command Line \(page 57\)](#)
- ♦ [iManager \(page 57\)](#)

Command Line

This command deletes the specified policy DN from the directory. It fails if the policy is in use by any principal.

To delete a policy, enter the following at the kadmin prompt:

```
delete_policy [-force] policy
```

For example, enter the following at the kadmin prompt:


```
delete_policy cn=realm-policy,o=org
```

You are prompted to confirm the deletion as follows:

```
Are you sure you want to delete the policy "cn=realm-policy,o=org"? (yes/no):
```

Enter “yes” to proceed with the deletion.

iManager

- 1 In Novell iManager, click the Roles and Tasks button .
- 2 Select Kerberos Management > Delete Password Policy.

Refer to the iManager online help for more information.

Viewing Policy Values

You can view the values of the specified policy as follows:

```
get_policy [-terse] policy
```

The `-terse` flag outputs the fields as quoted strings separated by tabs.

For example:

```
get_policy cn=realm-policy,o=org
```

This gives the following output:

```
Policy: cn=realm-policy,o=org
Maximum password life: 172800
Minimum password life: 86400
Minimum password length: 5
Minimum number of password character classes: 2
Number of old keys kept: 1
Reference count: 0
```

Listing Policies

You can list all the policy DN's as follows:

```
list_policies
```

This gives the following output:

```
cn=policy1,o=org
cn=realm-policy,o=org
```

Updating Kerberos LDAP Extension Information

You can update the `ldapExtensionInfo` attribute on the LDAP server object using the `kdb5_util` utility as follows:

```
kdb5_util [-D user_dn [-w passwd]] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]
```

```
ldapxtn_info -add|-clear [-t trusted_cert]
```

For example:

```
kdb5_util -D cn=admin,o=org -h ldap-server1.mit.edu -p 636 ldapxtn_info -add
```

Table 34 ldapxtn_info Parameter Description

Parameter	Description
-D	Distinguished name of the user who has sufficient rights to authenticate to the LDAP server.
-w	Specifies the userdn password. This is not recommended.
-h	Host name or IP Address of the server hosting LDAP service for a Kerberos realm.
-p	SSL port number of the LDAP server.

Parameter	Description
-t	Specifies the filename that contains Trusted Root Certificate of the LDAP server.
-add	Adds Kerberos LDAP extension information (OIDs for Kerberos LDAP Extensions) to ldapExtensionInfo on the LDAP server object.
-clear	Removes Kerberos LDAP extension information (OIDs for Kerberos LDAP Extensions) from ldapExtensionInfo on the LDAP server object.

Importing Trusted Root Certificate

To import a trusted root certificate, enter the following:

```
kdb5_util [-D user_dn [-w passwd]] [-h ldap_server] [-p ldap_port]
import_cert [-y] [-e encode_format] -f certificate_file
```

Table 35 import_cert Parameter Description

Parameter	Description
-D	Distinguished name of the user who has sufficient rights to authenticate to the LDAP server.
-w	Specifies the userdn password. We do not recommend you to use this.
-h	Host name or IP Address of the server hosting LDAP service for a Kerberos realm.
-p	SSL port number of the LDAP server.
-y	If specified, will not prompt the user to accept the certificate, instead assumes that user agrees to accept.
-e	Encoding format of the Trusted Root Certificate to be imported. DER is the default encoding format used.
-f	Complete path of the file which will contain the Trusted Root Certificate upon successfully getting the certificate.

For example:

```
kdb5_util -h ldap-server1.mit.edu -p 636 import_cert -e der -f /opt/novell/
kerberos/certs/trustedroot.der
```

The output is as follows:

```
Trusted Root Certificate received.
Certificate Information:
  Status:      self signed certificate in certificate chain
  Issuer:      /OU=Organizational CA/O=KAILASA
  Subject:     /OU=Organizational CA/O=KAILASA
  Valid From:  Saturday, April 02, 2005 03:18:56 PM IST
  Valid Till:  Thursday, April 02, 2015 03:18:56 PM IST
```

```
Would you like to accept the certificate? (Y/N): y
```

Setting the Master Key

If the master key of a realm in eDirectory is corrupted, you can reset it using `kdb5_util`. Ensure that the master key is reset with the same master password and key type, which was provided while creating the realm. Else, all the principals in the realm will be unusable.

If you change the master key of a realm, then the existing principals will not be able to access any Kerberos services in the network, as their secret keys were encrypted with the old master key. If you want to reset the master key, you have to delete and reset the keys for all the principals in the realm.

You can reset the master key as follows:

```
kdb5_util [-D user_dn [-w passwd]] [-h ldap_server]
          [-p ldap_port] [-t trusted_cert]

setmasterkey [-k mkeytype] [-m|-P password] [-r realm]
```

For example:

```
kdb5_util -D cn=admin,o=org -h ldap-server1.mit.edu -p 636 setmasterkey -r
ATHENA.MIT.EDU
```

Table 36 **setmasterkey Parameter Description**

Parameter	Description
-D	Distinguished name of the user who has sufficient rights to authenticate to the LDAP server.
-w	Specifies the userdn password. We do not recommend you to use this.
-h	Host name or IP Address of the server hosting LDAP service for a Kerberos realm.
-p	SSL port number of the LDAP server.
-t	Specifies the filename that contains Trusted Root Certificate of the LDAP server.
-k	Specifies the key type of the master key for the realm; if not specified, the default value is used. The default value is DES3_HMAC_SHA1.
-m	Specifies that the master password should be read from the keyboard.
-P	Specifies the master password. We do not recommend you to use this.
-r	Specifies the Kerberos realm of the database; by default the default_realm parameter of configuration file (/etc/krb5.conf) is used.

Changing Principal Password

The end users can change their own passwords using the `kpasswd` utility and the syntax is as follows:

```
kpasswd principal_name
```

You will be prompted to enter your existing password and new password. The `kpasswd` utility uses the Kerberos Password Server for changing the Kerberos passwords.

4 Integrating Universal Password

Universal password is a single password to eDirectory. It allows synchronization of passwords from eDirectory to other systems. For more information about universal passwords, refer to *Deploying Universal Password* (<http://www.novell.com/documentation/nmas23/admin/data/allq21t.html>).

Novell Kerberos KDC can be integrated with universal password so that there is a single password to authenticate to eDirectory and Kerberos. The eDirectory and Kerberos password are synchronized using the Kerberos Password Agent.

Configuring Universal Passwords

Prerequisites

- Enable universal password in eDirectory.

You can enable universal password at tree, container, or user level.

For more information, refer to *Deploying Universal Password* (<http://www.novell.com/documentation/nmas23/admin/data/allq21t.html>)

- Ensure that the “Synchronize Distribution Password while setting universal password” option is enabled in the password policy at the level (tree, container, or user) that is in effect.

Integrating Universal Password with Novell Kerberos KDC

To integrate universal password, complete the following procedure:

- 1** [Enabling Universal Passwords \(page 61\)](#)
- 2** [Setting or Modifying the Universal Passwords \(page 63\)](#)

Enabling Universal Passwords

In Novell Kerberos KDC, you can enable universal password at the realm or user level.

Enabling Universal Passwords at the Realm Level

Enable universal password at the time of creating the realm. Alternatively, once the realm is created, you can enable universal passwords by editing the realm.

You can enable universal passwords through any of the following methods:


Command Line

```
kdb5_util -h ldap-server1.mit.edu -D cn=admin,o=org -r ATHENA.MIT.EDU create  
-subtree o=org -up
```

```
kdb5_util -h ldap-server1.mit.edu -D cn=admin,o=org -r ATHENA.MIT.EDU modify
-up
```

NOTE: To disable universal password, use the above command with the `-clearup` option.

iManager

1 In Novell iManager, click the Roles and Tasks button .

2 Click Kerberos Management > New Realm.

If you are modifying the realm, click Kerberos Management > Edit Realm.

Refer to the iManager online help for more information.

Enabling Universal Passwords at the User Level


Enable universal password for the principal at the time of creating the principal. Alternatively, once the principal is created, you can enable universal passwords by editing the principal.

Command Line

```
add_principal -x up=on -x userdn=cn=admin,o=org testing
modify_principal -x up=on testing
```

NOTE: To disable universal password, use the above commands with `up=off` or `up=clr` options.

iManager

1 In Novell iManager, click the Roles and Tasks button .

2 Select Kerberos Management > New Principal.

If you are modifying the realm, click Kerberos Management > Edit Principal.

Refer to the iManager online help for more information.

When universal password is enabled or disabled for a principal, it will be applicable to all the principals of the user object with which the principal is associated.

The table can be used to check if universal password has been enabled for a user.

Table 37 Universal Password Enabled or Disabled

Universal Password Configuration Level		Is Universal Password Enabled?
Realm	User	
True	True	Yes
True	False	No
False	True	Yes
False	False	No
True	Not Present	Yes
False	Not Present	No
Not Present	True	Yes

Universal Password Configuration Level		Is Universal Password Enabled?
Not Present	False	No
Not Present	Not Present	No

Ensure that Kerberos Password Agent is loaded when universal password is enabled in Kerberos and eDirectory. If the Kerberos Password Agent is not running, then the passwords will not get synchronized when the universal password is changed in eDirectory. Additionally, when the password is changed using cpw or kpasswd in Kerberos, the principal's Kerberos key version might not be consistent.

Setting or Modifying the Universal Passwords

For the universal password to be effective, it needs to be either set (new users) or modified (existing users). You can set or modify the universal password either in eDirectory or Kerberos.

In Kerberos, the universal password is set when a new user is created while adding a new principal. If the principal is added for an already existing user, then the universal password is not set but only the principal's Kerberos keys are set. However, the user can be asked to change the password (universal password) to synchronize universal password and Kerberos passwords.

In Kerberos, when the Kerberos password is changed using kpasswd or cpw, it changes the universal password.

Kerberos Password Agent

The Kerberos Password Agent (KPA) synchronizes the Kerberos password with universal password based on the configuration at the realm and user. KPA must be installed on all the eDirectory servers with writable replica of the Kerberos data that the users use to change passwords.

To start KPA, enter the following:

```
kpa -l
```

To stop KPA, enter the following:

```
kpa -u
```

The messages logged by the Password Agent will be displayed when the Misc tag is enabled in the ndstrace. In eDirectory 8.8, the messages are also logged in the log file that is configured.

WARNING: The Kerberos Password Agent is not loaded automatically when the machine or eDirectory is restarted. It has to be loaded manually.

Key Generation

The encryption types and salt type used by the Kerberos Password Agent to generate the Kerberos keys from the universal password is based on the following:

- ◆ If the principal has Kerberos keys, the encryption and salt types used in generating the existing keys will be used to generate the new keys from the universal password.
- ◆ If the principal does not have the Kerberos password set, the realm configuration is used to determine the encryption and salt types to be used for key generation.

- ◆ The default encryption type or supported encryption types (configured at the realm) is used, with default type taking precedence over supported. If both these values are not configured, the encryption type used is DES3-HMAC-SHA1.
- ◆ Similarly, for the salt type, the default salt type or supported salt types (configured at the realm) is used, with default type taking precedence over supported. If both these values are not configured, the salt type used is NORMAL.

The following table illustrates some of the encryption and salt type combinations for the key generation:

Table 38 Key Generation Logic

Encryption Type		Salt Type		Key Generated
Default (Defined during realm configuration)	Supported (Defined during realm configuration)	Default (Defined during realm configuration)	Supported (Defined during realm configuration)	
✓	✓	✓	✓	One key with default encryption type and default salt type.
✗	✓	✗	✓	Multiple keys with all the supported encryption types and supported salt types combinations.
✗	✗	✗	✗	One key with the DES3-HMAC-SHA1 encryption type and NORMAL salt type.
✓	✗	✗	✓	Multiple keys with the default encryption type and all the supported salt types combinations.
✗	✓	✓	✗	Multiple keys with the all the supported encryption types and the default salt types combinations.

For more information on the supported encryption and salt types, refer to [Appendix B, “Supported Encryption Types and Salt Types,”](#) on page 85.

Universal Password Considerations

- ◆ If the universal password is enabled, the randkey option cannot be used for setting universal password while creating a principal or changing the password of a principal.
- ◆ Setting the universal password for a principal associated with a user object sets universal password as the Kerberos password for all the principals associated with that user object.
- ◆ If universal password is enabled, the Kerberos Password Agent module should be loaded whenever machine or the eDirectory is restarted.

- ◆ Novell Kerberos KDC does not support extended characters in password. If the Kerberos password is integrated with universal password, the universal password also cannot have extended characters.
- ◆ While bulkloading user principals with universal passwords, there will be a degradation in the performance, as the Kerberos keys are synchronized at the time of the creation of these principals.

5

Deployment Notes

This section gives some of notes you can follow at the time of deploying Novell® Kerberos KDC.

- ♦ “Optimizing the Performance” on page 67
- ♦ “LDAP Connection Pool” on page 67
- ♦ “Bulkloading Principals” on page 69

Optimizing the Performance

The LDAP search performance on principal name attributes impacts the authentication performance of Novell Kerberos KDC.

Before deploying Novell Kerberos KDC, review and consider the following guidelines to optimize the LDAP search:

- ♦ **Create a DS replica indexed on the krbPrincipalName attribute:** During Kerberos authentication by Novell Kerberos KDC, it searches for the principal name attribute within the specified sub-tree containers. The search is faster when the container is small and flat. The search time increases as the size and nesting increase.

To increase the search performance, create separate DS replicas and implement value indexing on the krbPrincipalName attribute. Use this replica server as the LDAP server for KDC, Administration, and Password server access. This indexing on the principal name improves the speed of the search.

- ♦ **Create aliases for identities in large trees:** If a large eDirectory tree has limited users with Kerberos identities spread all over the tree, we recommend creating Kerberos alias objects for those eDirectory users and keeping all the Kerberos alias objects under the realm container. This simplifies the search to a great extent and increases the speed of the Kerberos authentication performance.

LDAP Connection Pool

Novell® Kerberos KDC uses LDAP to access eDirectory™. This means that whenever the eDirectory or LDAP services are down or are restarted for maintenance purpose, the Novell Kerberos KDC services get affected. Additionally, the Novell Kerberos KDC services need to be restarted manually whenever the eDirectory or LDAP services are restored.

Novell Kerberos KDC provides a mechanism to overcome this problem as follows:

- ♦ Establishes LDAP connections with multiple LDAP servers.

If any of the server is not responding, the LDAP connections with the other servers are utilized. This means that if all the LDAP servers are down, the Novell Kerberos KDC services will not abort, but will handle the requests appropriately, by returning an error. Whenever any

of the LDAP server is restored, the LDAP module attempts to reconnect with all the LDAP servers until it gets a connection.

- ◆ Enables multiple simultaneous client requests to access the database functionality.

As multiple LDAP connections are cached for every LDAP server, multiple requests from the Novell Kerberos KDC services are serviced simultaneously.

The list of LDAP servers and number of connections per server can be set in `/etc/krb5.conf` file.

Configuring LDAP Connection Pool

To configure LDAP connection, you need to set the following:

- ◆ [Setting the LDAP Servers List \(page 68\)](#)
- ◆ [Setting Number of Connections Per Server \(page 68\)](#)

Novell Kerberos KDC services read the database-specific parameters from the `/etc/krb5.conf` configuration file. You can provide these parameters at the command line too. This helps the administrator to avoid frequent modification of the configuration file and to modify the options even without write permissions on the configuration file. Additionally, many server requests with different parameter values on a single machine are also possible.

Setting the LDAP Servers List

You can set up the LDAP servers using any of the following methods:

The list of the LDAP servers that the Novell Kerberos KDC server tries to connect is defined by the `ldap_servers` parameter in the `/etc/krb5.conf` file.

- ◆ **Configuration File**

Use the `ldap_servers` parameter in the `/etc/krb5.conf` file as follows:

```
ldap_servers = ldap-server1.mit.edu ldap-server2.mit.edu:1636
```

- ◆ **Command Line**

Use the following command line option to set the list of LDAP servers that the Kerberos service (KDC, Administration, and Password) should connect to.

```
-x host=hostname:port
```

Setting Number of Connections Per Server

If a Kerberos Service, such as KDC, consumes the Database service from multiple LDAP servers then the attribute `ldap_conns_per_server` in the `/etc/krb5.conf` is set to an optimum value so that the database operation load is distributed to multiple servers.

Multiple secure (SSL) connections can be established with every LDAP server on need basis.

You can set up the number of LDAP connections per server using any of the following methods:

- ◆ **Configuration File**

Use the `ldap_conns_per_server` parameter in the `/etc/krb5.conf` file as follows:

```
ldap_conns_per_server = 5
```

- ◆ **Command Line**

Use the following command line option to limit the number of LDAP connections that the Kerberos service (KDC, Administration, and Password) should use:

```
-x nconns=value
```

Bulkloading Principals

While bulkloading user principals with LDIF files, include `krbPrincipalAux`, `krbPolicyAux`, and `krbPwdPolicyRefAux` in `objectClass`.

For example,

```
version: 1

dn: cn=jsmith,ou=engineering,o=acme
changetype: add
objectclass: User
objectclass: krbPrincipalAux
objectclass: krbPolicyAux
objectclass: krbPwdPolicyRefAux
cn: jsmith
Surname: smith
krbPrincipalName: jsmith@ACME.COM
```

Though none of the attributes in the object classes `krbPolicyAux` and `krbPwdPolicyRefAux` are specified in the LDIF file for the creation of the user principals, failing to include these object classes will make the Kerberos administration and `kpasswd` utilities fail, as they refer to these classes.

6

Interoperability with MIT and Microsoft KDCs

This section describes how to achieve interoperability between MIT and Microsoft* KDCs cross-realm setup as follows:

- ♦ [“Interoperability with MIT KDC” on page 71](#)
- ♦ [“Interoperability with Microsoft KDC” on page 72](#)
- ♦ [“How Cross-Realm Setup Works” on page 74](#)

The procedures use the following terminology:

- ♦ MIT realm: mitrealm
- ♦ Win2K domain: w2kdomain
- ♦ Novell Kerberos KDC realm: novlrealm

Replace the realm names specified above with the names chosen by the Kerberos administrator.

Interoperability with MIT KDC

- ♦ [“Accessing Services in mitrealm from novlrealm” on page 71](#)
- ♦ [“Accessing Services in novlrealm from mitrealm” on page 71](#)

Accessing Services in mitrealm from novlrealm

To access services, set up a cross-realm authentication between *novlrealm* and *mitrealm* as follows:

- 1** In *novlrealm*, create a principal, *krbtgt/mitrealm@novlrealm*.
- 2** In *mitrealm*, create a principal, *krbtgt/mitrealm@novlrealm*.
- 3** In the appropriate Kerberos configuration file (*/etc/krb5.conf*), create entries for *novlrealm* and *mitrealm*.

IMPORTANT: Make sure that in both realms the password or key of the *krbtgt/mitrealm@novlrealm* is same.

Accessing Services in novlrealm from mitrealm

To access services, set up cross-realm authentication between *novlrealm* and *mitrealm*:

- 1** In *mitrealm*, create a principal, *krbtgt/novlrealm@mitrealm* .
- 2** In *novlrealm*, create a principal, *krbtgt/novlrealm@mitrealm* .
- 3** In the appropriate Kerberos configuration file (*/etc/krb5.conf*), create entries for *novlrealm* and *mitrealm*.

IMPORTANT: Make sure that in both realms the password or key of the *krbtgt/novlrealm@mitrealm* is same.

Interoperability with Microsoft KDC

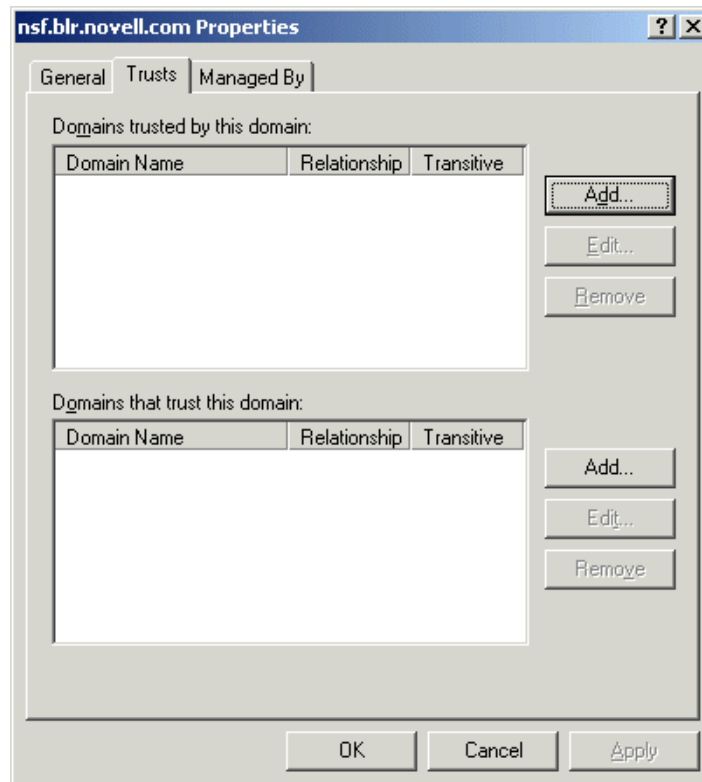
Accessing Services in w2kdomain from novlrealm

To set up cross-realm authentication between *novlrealm* and *w2kdomain*:

- 1** (Conditional) If a user object does not already exist for a user in Active Directory, then create a user object

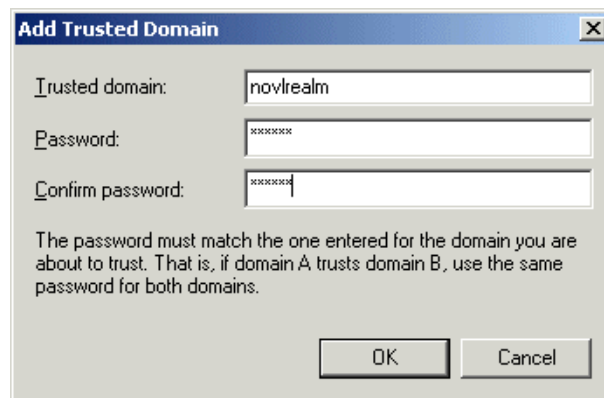
User creation is required in order to get tickets containing PAC (authorization data honored by application services in w2kdomain) from Microsoft Active Directory or KDC.
- 2** Map the user's principal in *novlrealm* to this user object:
 - 2a** Click Start > Programs > Administrative Tools > Active Directory Users and Computers.
 - 2b** Right-click the user object > Name Mappings.
 - 2c** Click Kerberos Names tab > Add.
 - 2d** Enter the user's principal name.
- 3** Set up a trust between *w2kdomain* and *novlrealm*:
 - 3a** Click Start > Programs > Administrative Tools > Active Directory Domains and Trusts
 - 3b** Click win2kdomain > Properties > Trusts
 - 3c** Click Add in the Domains trusted by this domain pane (as in figure 3) to display the Add Trusted dialog box.

Figure 4 Accessing Services in w2kdomain from novrealm



3d In the Add Trusted Domain dialog box, enter *novrealm* as the trusted domain.

Figure 5 Adding Trusted Domain



3e Enter the password and reenter it to confirm the password.

IMPORTANT: Make sure that in both realms the password or key of *krbtgt/w2kdomain@novrealm* is the same.

3f Click OK to ignore the warning message about non-Windows Kerberos realm.

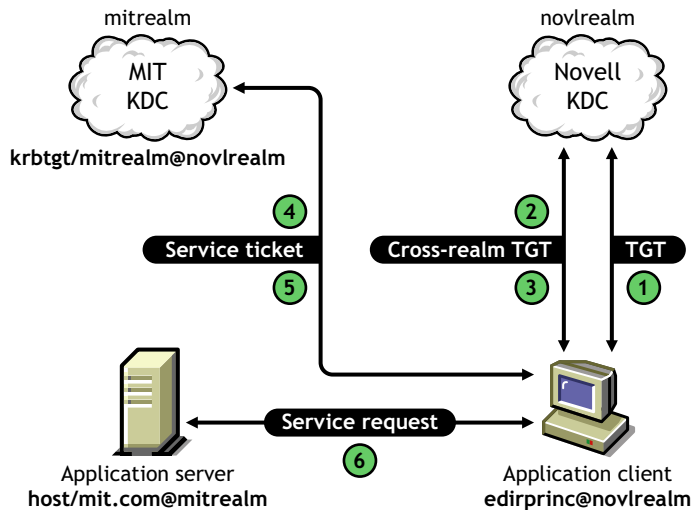
4 In *novrealm*, create a principal, *krbtgt/w2kdomain@novrealm*.

5 In the appropriate Kerberos configuration file (*/etc/krb5.conf*), create entries for *novrealm* and *mitrealm*.

How Cross-Realm Setup Works

Figure 5 uses the example of accessing a service in the MIT KDC realm from a KDC realm.

Figure 6 Cross-realm setup working



The activity listed below uses the following terminology:

eDirectory user: *ediruser.novell*

User principal: *edirprinc@novrealm*

Service principal: *host/mit.com@mitrealm*

The background activity in a cross-realm setup is explained below:

1. An eDirectory user authenticates to **novrealm** as *edirprinc@novrealm*.
2. The application client requests a service ticket for the principal, *host/mit.com@mitrealm* from KDC Server (hosting **novrealm**).
3. The KDC Server sends a service ticket for the principal, *krbtgt/mitrealm@novrealm* to the client.
4. The client sends this cross-realm ticket to MIT KDC (hosting **mitrealm**) along with a request for a service ticket for the principal, *host/mit.com@mitrealm*.
5. MIT KDC sends the service ticket for *host/mit.com@mitrealm* to the application client.
6. The client sends this service ticket to the application server.

7

Security Considerations

This chapter provides information on the security considerations of Novell® Kerberos KDC:

1. Use **SSL mutual authentication** or **SASL EXTERNAL bind** for authenticating the Kerberos services.
2. Secure the connection between your Web browser and the iManager server with SSL and the connection between iManager and Novell eDirectory™. Failing to do so will cause the Kerberos sensitive data like master key and principal key to be sniffed during the creation of the realm and principals.
3. Protect the following files with appropriate file system rights:
 - ◆ Configuration file (/etc/krb5.conf)
 - ◆ Service password stash file (specified with the `ldap_service_password_file` parameter in /etc/krb5.conf)
 - ◆ ACL file for administration (specified with the `acl_file` parameter in /etc/krb5.conf)
 - ◆ Password dictionary file (specified with the `dict_file` parameter in /etc/krb5.conf)
 - ◆ Certificate files of Kerberos service.
 - ◆ Trusted root certificates of the LDAP servers (specified with the `ldap_root_certificate_file` parameter in /etc/krb5.conf)
 - ◆ Log files of KDC, Administration, and Password servers, as these contain auditing information.
 - ◆ Kerberos keytab files (default location is /etc/krb5.keytab)

All these files must be stored only on the local storage device and not on remotely mounted devices. The recommended file permissions for these files are RW for root. Additionally, protect these files during backup and restore operations.

4. Use the strongest cryptographic algorithm for the master and principal keys. Use DES and RC4 only for interoperability with other Kerberos distributions.
5. Keep the Kerberos servers in a physically secure location with the access only to the authorized personnel.
6. TGS (`krbtgt/REALM@REALM`), Administration service (`kadmin/admin@REALM`), and Password service (`kadmin/changepw@REALM`) principal keys must be randomly generated and periodically reset.

IMPORTANT: We do not recommend the use of Administration server, as it needs almost the supervisor rights. Instead, we recommend using `kadmin.local` that directly communicates with eDirectory using LDAP over SSL. We also recommend you to use the Novell Kerberos KDC iManager plug-ins.

8

Troubleshooting Kerberos KDC

This section provides troubleshooting information that you can use to resolve some of the commonly faced problems while using Novell® Kerberos KDC.

- ◆ [“Installation” on page 77](#)
- ◆ [“Starting the Services” on page 77](#)
- ◆ [“KDC” on page 78](#)
- ◆ [“kdb5_util” on page 78](#)
- ◆ [“kadmin” on page 80](#)
- ◆ [“Password Agent” on page 80](#)
- ◆ [“iManager Plug-in” on page 81](#)

Installation

This section provides information about the following troubleshooting scenarios:

- ◆ [“Installation fails” on page 77](#)
- ◆ [“Installation fails randomly to add some of the packages on NLD” on page 77](#)

Installation fails

Possible Cause: MIT Kerberos may already be installed on the system. The location of the startup scripts is the same for both implementations.

Action: Manually uninstall the MIT KDC and then install Novell Kerberos KDC on the system.

Installation fails randomly to add some of the packages on NLD

Action: Retry after sometime it will install the packages on NLD.

Starting the Services

This sections explains the problems encountered while starting the KDC, Administration server, or Password server.

- ◆ [“Server stops functioning with the error message "File Size Exceeded"” on page 78](#)
- ◆ [“Server fails to start with the error message “master key read failed : \[-603\] - while fetching master key K/M for realm”” on page 78](#)
- ◆ [“Server fails to start with error message "Invalid credentials"” on page 78](#)

Server stops functioning with the error message "File Size Exceeded"

Possible Cause: Log file size cannot exceed 2 GB.

Action: To restart KDC, delete the log file manually. To avoid this situation, we recommend you to back up and truncate the log file on a regular basis.

Server fails to start with the error message "master key read failed : [-603] - while fetching master key K/M for realm"

Possible Cause: The Service object is not assigned with rights on the realm container.

Action: Modify the realm by specifying that the corresponding Service object pertains to this realm as follows:

```
kdb5_util -D cn=admin,o=org modify_service -realm ATHENA.MIT.EDU
cn=service-kdc,o=org
```

NOTE: For more information, refer to "Modifying a Service" on page 38.

Server fails to start with error message "Invalid credentials"

Possible Cause: Problem in stashing service passwords.

Action: Check if the entry corresponding to the service object is proper in the stash file. If there are multiple entries (possibly due to case errors in input), delete all entries except the last one. Then, restart the server.

KDC

This section provides information about the following troubleshooting scenarios:

- ◆ "KDC fails to start with error message "krb5kdc: master key type mismatch"" on page 78

KDC fails to start with error message "krb5kdc: master key type mismatch"

Possible Cause:

1. The master key type specified with the -k command line option does not match with the one in eDirectory.
2. Realm was created with the default value for the master key type but the -k command line option with appropriate type was not specified while invoking the KDC.

Action: Specify the correct master key type with -k option. If you have created the realm with the default master key type, specify DES3-HMAC-SHA1 as the value for -k option.

kdb5_util

This section provides information about the following troubleshooting scenarios:

- ◆ "On creating a realm the error message displayed is create: Realm creation FAILED:[2] Set Master key failed while creating realm 'ATHENA.MIT.EDU'" on page 79
- ◆ "create_service or setsrvpw commands fail to set the service object password with the error message, "FAILED: DSA is unwilling to perform. Failed to set password for service object"" on page 79

- ♦ “On adding principal the error message displayed is `add_principal: FAILED: Insufficient access`” on page 79
- ♦ “User is unable to create a principal” on page 79
- ♦ “Utility fails with Constraint Violation error” on page 79
- ♦ “On destroying a realm the error message displayed is `destroy: DN information missing deleting database of 'ATHENA.MIT.EDU'`” on page 80

On creating a realm the error message displayed is

create: Realm creation FAILED:[2] Set Master key failed while creating realm 'ATHENA.MIT.EDU'

Possible Cause: LDAP extension not added

Action: Add LDAP extension using `kdb5_util`. For example,

```
kdb5_util -D cn=admin,o=org -w secret ldapxtn_info -add
```

Possible Cause: LDAP extension not loaded

Action: Load LDAP extension by restarting the LDAP server

create_service or setsrvpw commands fail to set the service object password with the error message, “FAILED: DSA is unwilling to perform. Failed to set password for service object”

Possible Cause: The password may be violating the password policy configured for the container in the which the service object is created.

Action: If the password is specified manually, ensure that the password adheres to the policy that is configured.

If `-randpw` option is used, ensure that the password policy allows a password of 128 characters.

On adding principal the error message displayed is `add_principal: FAILED: Insufficient access`

Possible Cause: Admin service object is not assigned with sufficient rights over the realm container.

Action: Create admin service object as follows:

```
kdb5_util - D cn=admin,o=org create_service -admin - realm
ATHENA.MIT.EDU - randpw - f /home/andrew/conf_keyfile cn=admin-
service,o=org
```

NOTE: Ensure that the administrative service object dn is specified in the `krb5.conf` file.

User is unable to create a principal

Possible Cause: Universal password is enabled for Novell Kerberos KDC and not enabled for eDirectory

Action: Either enable universal password for eDirectory or create a principal without enabling universal password.

Utility fails with Constraint Violation error

Possible Cause: Constraint Violation error can occur due to several reasons:

1. Adding attribute with no value when at least one is required.
2. Adding attribute with many values when only one is allowed.

3. Adding attribute value that conflicts with the syntax.
4. Exceeding the size boundary.

Action: Check the attribute values, syntax, and size.

On destroying a realm the error message displayed is destroy: DN information missing deleting database of 'ATHENA.MIT.EDU'

Possible Cause: `ldap_conns_per_server` parameter value in `krb5.conf` file is less than 3

Action: Set the `ldap_conns_per_server` parameter value to 3 or above

kadmin

This section provides information about the following troubleshooting sections:

- ◆ [“Unable to create a Kerberos user principal” on page 80](#)
- ◆ [“Utility fails with Constraint Violation error” on page 80](#)

Unable to create a Kerberos user principal

Possible Cause: `userdn` is more than 64 characters.

Action: Restrict the `userdn` to 64 characters.

Utility fails with Constraint Violation error

Possible Cause: Constraint Violation error can occur due to several reasons:

1. Adding attribute with no value when at least one is required.
2. Adding attribute with many values when only one is allowed.
3. Adding attribute value that conflicts with the syntax.
4. Exceeding the size boundary.

Action: Check the attribute values, syntax, and size.

Password Agent

This section provides information about the following troubleshooting scenarios:

- ◆ [“Error messages in the password agent log file for universal password” on page 80](#)

Error messages in the password agent log file for universal password

Possible Cause: In eDirectory 8.8, when the user's universal password is set for the first time, the Password Agent log file displays the following error messages:

```
KPA: Failed to read the Universal Password of user [Error: -1658]
KPA: Failed to update the Kerberos keys of user followed by success messages
  if the loglevel is enabled to display informational messages
KPA: Kerberos password modified for principal
KPA: Principals of user successfully updated with Universal Password
```

Explanation: The Password Agent again reads the user's Universal Password and updates the Kerberos keys with Universal Password.

Action: Ignore the error messages.

iManager Plug-in

This section provides information about the following troubleshooting scenarios:

- ◆ “User is unable to create a principal” on page 81
- ◆ “Failure in schema extension failed or realm creation” on page 81
- ◆ “System Error while performing the tasks” on page 81

User is unable to create a principal

Possible Cause: Universal password is enabled for Novell Kerberos KDC and not enabled for eDirectory.

Action: Either enable universal password for eDirectory or create a principal without enabling universal password.

Failure in schema extension failed or realm creation

Possible Cause: Trusted Root Certificate is not imported into the keystore.

Action: Import the trusted root certificate from the LDAP server into the keystore and restart the Tomcat server.

System Error while performing the tasks

Possible Cause: iManager version is lesser than 2.5.

Action: Install the Novell Kerberos KDC plug-ins on iManager 2.5.

A

Sample krb5.conf File

A sample krb5.conf file is provided in the *untarred_path*/NovellKerberosKDC/setup directory. You can use the /etc/krb5.conf configuration file to set the default values. While managing Novell Kerberos KDC, when you do not specify any of the mandatory parameters, the values are taken from the /etc/krb5.conf file. This file looks similar to the following:

```
[libdefaults]
default_realm = ATHENA.MIT.EDU

[realms]
  ATHENA.MIT.EDU = {
    max_life = 10h 0m 0s
    max_renewable_life = 7d 0h 0m 0s
    acl_file = /opt/novell/kerberos/kadm5.acl
    dict_file = /opt/novell/kerberos/kadm5.dict
    kdc = kerberos.mit.edu
    admin_server = kerberos-1.mit.edu
    kpasswd_server = kerberos-1.mit.edu
    database_module = ldapconf
  }

[kdcdefaults]
num_threads = 10

[domain_realm]
.mit.edu = ATHENA.MIT.EDU
mit.edu = ATHENA.MIT.EDU

[logging]
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log
kpasswd_server = FILE:/var/log/kpasswdd.log

[dbdefaults]
database_module = ldapconf

[dbmodules]
  ldapconf = {
    db_library = kdb_ldap
    ldap_ssl_port = 636
    ldap_kdc_dn = "cn=KDC Server - kerberos.mit.edu,o=mit"
    ldap_kadmind_dn = "cn=Admin Server - kerberos.mit.edu,o=mit"
    ldap_kpasswdd_dn = "cn=Passwd Server - kerberos.mit.edu,o=mit"
    ldap_root_certificate_file = /opt/novell/kerberos/TrustedRoot-
      ldap-server1.mit.edu.der /opt/novell/kerberos/TrustedRoot-ldap-
      -server2.mit.edu.der
    ldap_service_password_file = /opt/novell/kerberos/keyfile
    realm_read_refresh_interval = 300
    ldap_servers = ldap-server1.mit.edu ldap-server2.mit.edu:1636
    ldap_conns_per_server = 5
```

}

B Supported Encryption Types and Salt Types

This chapter lists the supported encryption and salt types supported in Novell® Kerberos KDC.

Supported Encryption Types

The following encryption types are supported by the Novell Kerberos KDC components

- ◆ des-cbc-crc: DES cbc mode with CRC-32
- ◆ des-cbc-md4: DES cbc mode with RSA-MD4
- ◆ des-cbc-md5: DES cbc mode with RSA-MD5
- ◆ des3-cbc-sha1
- ◆ des3-hmac-sha1
- ◆ des3-cbc-sha1-kd: triple DES cbc mode with HMAC/sha1
- ◆ aes256-cts-hmac-sha1-96
- ◆ aes256-cts: AES-256 CTS mode with 96-bit SHA-1 HMAC
- ◆ aes128-cts-hmac-sha1-96
- ◆ aes128-cts: AES-128 CTS mode with 96-bit SHA-1 HMAC
- ◆ arcfour-hmac
- ◆ rc4-hmac
- ◆ arcfour-hmac-md5: RC4 with HMAC/MD5

Supported Salt Types

Your Kerberos key is derived from your password. To ensure that users who happen to have the same password do not have the same key, Kerberos 5 incorporates more information into the key using something called a salt. The supported values for salts are as follows.

- ◆ normal: default for Kerberos Version 5
- ◆ v4: the only type used by Kerberos Version 4, no salt
- ◆ norealm: same as the default, without using realm information
- ◆ onlyrealm: uses only realm information as the salt
- ◆ special: only used in very special cases; not fully supported

C

Administrative Privileges for Kerberos Database

You need to set administrative privileges for the Kerberos database are stored in the file `kadm5.acl`.

The format of the file is:

```
Kerberos_principal permissions [target_principal] [restrictions]
```

Table 39 `kadm5.acl` File

Field	Description																								
Kerberos_principal	<p>The Kerberos principal (and optional target principal) can include the "*" wildcard, so if you want any principal with the instance "admin" to have full permissions on the database, you could use the principal <code>*/admin@REALM</code> where "REALM" is your Kerberos realm.</p> <p>NOTE: Common use of an admin instance is so you can grant separate permissions (such as administrator access to the Kerberos database) to a separate Kerberos principal. For example, the user <i>joeadmin</i> might have a principal for his administrative use, called <i>joeadmin/admin</i>. This way, <i>joeadmin</i> would obtain <i>joeadmin/admin</i> tickets only when he actually needs to use those permissions.</p>																								
target_principal	Can also include backreferences to Kerberos_principal, in which "*"number" matches the component number in the Kerberos_principal.																								
permissions	<p>The permissions are represented by single letters; UPPER-CASE letters represent negative permissions. The permissions are:</p> <table border="1"> <tbody> <tr> <td>a</td> <td>Allows the addition of principals or policies in the database.</td> </tr> <tr> <td>A</td> <td>Disallows the addition of principals or policies in the database.</td> </tr> <tr> <td>d</td> <td>Allows the deletion of principals or policies in the database.</td> </tr> <tr> <td>D</td> <td>Disallows the deletion of principals or policies in the database.</td> </tr> <tr> <td>m</td> <td>Allows the modification of principals or policies in the database.</td> </tr> <tr> <td>M</td> <td>Disallows the modification of principals or policies in the database.</td> </tr> <tr> <td>c</td> <td>Allows the changing of passwords for principals in the database.</td> </tr> <tr> <td>C</td> <td>Disallows the changing of passwords for principals in the database.</td> </tr> <tr> <td>i</td> <td>Allows inquiries to the database.</td> </tr> <tr> <td>I</td> <td>Disallows inquiries to the database.</td> </tr> <tr> <td>l</td> <td>Allows the listing of principals or policies in the database.</td> </tr> <tr> <td>L</td> <td>Disallows the listing of principals or policies in the database.</td> </tr> </tbody> </table>	a	Allows the addition of principals or policies in the database.	A	Disallows the addition of principals or policies in the database.	d	Allows the deletion of principals or policies in the database.	D	Disallows the deletion of principals or policies in the database.	m	Allows the modification of principals or policies in the database.	M	Disallows the modification of principals or policies in the database.	c	Allows the changing of passwords for principals in the database.	C	Disallows the changing of passwords for principals in the database.	i	Allows inquiries to the database.	I	Disallows inquiries to the database.	l	Allows the listing of principals or policies in the database.	L	Disallows the listing of principals or policies in the database.
a	Allows the addition of principals or policies in the database.																								
A	Disallows the addition of principals or policies in the database.																								
d	Allows the deletion of principals or policies in the database.																								
D	Disallows the deletion of principals or policies in the database.																								
m	Allows the modification of principals or policies in the database.																								
M	Disallows the modification of principals or policies in the database.																								
c	Allows the changing of passwords for principals in the database.																								
C	Disallows the changing of passwords for principals in the database.																								
i	Allows inquiries to the database.																								
I	Disallows inquiries to the database.																								
l	Allows the listing of principals or policies in the database.																								
L	Disallows the listing of principals or policies in the database.																								

Field	Description
s	Allows the explicit setting of the key for a principal.
S	Disallows the explicit setting of the key for a principal.
*	All privileges (admcil).
X	All privileges (admcil); identical to "*" .
restrictions	The restrictions are a string of flags. Allowed restrictions are:
[+ -]flagname	Flag is forced to indicated value. The permissible flags are the same as the + and - flags for the kadmin addprinc and modprinc commands.
-clearpolicy	Policy is forced to clear.
-policy pol	Policy is forced to be pol.
expire time	
pwexpire time	
maxlife time	
maxrenewlife time	Associated value will be forced to MIN (time, requested value).

The above flags act as restrictions on any add or modify operation which is allowed due to that ACL line.

An example of a kadm5.acl file:

NOTE: Note that order is important; permissions are determined by the first matching entry.

```
*/admin@ATHENA.MIT.EDU *
joadmin@ATHENA.MIT.EDU ADMCIL
joadmin/*@ATHENA.MIT.EDU il */root@ATHENA.MIT.EDU
*@ATHENA.MIT.EDU cil *1/admin@ATHENA.MIT.EDU
*/*@ATHENA.MIT.EDU i
*/admin@EXAMPLE.COM * -maxlife 9h -postdateable
```

In the above file, any principal in the ATHENA.MIT.EDU realm with an admin instance has all administrative privileges.

The user joadmin has all permissions with his admin instance, joadmin/admin@ATHENA.MIT.EDU (matches the first line).

He has no permissions at all with his null instance, joadmin@ATHENA.MIT.EDU (matches the second line). His root instance has inquire and list permissions with any other principal that has the instance root.

Any principal in ATHENA.MIT.EDU can inquire, list, or change the password of their admin instance, but not any other admin instance. Any principal in the realm ATHENA.MIT.EDU (except for joadmin@ATHENA.MIT.EDU, as mentioned above) has inquire privileges.

Finally, any principal with an admin instance in EXAMPLE.COM has all permissions, but any principal that they create or modify will not be able to get postdateable tickets or tickets with a life of longer than 9 hours.