Command Reference

Version 4.1
Contents

Preface .......................................................... vii
Who should read this book ........................................ vii
What this book contains .......................................... vii
Publications ....................................................... viii
  Release information ............................................. viii
  Base information .............................................. viii
  WebSEAL information .......................................... viii
  Web security information ....................................... viii
  Developer references .......................................... ix
  Technical supplements .......................................... ix
  Related publications ............................................ ix
  Accessing publications online ................................. xi
Accessibility ...................................................... xi
Contacting software support .................................... xi
Conventions used in this book .................................... xii
  Typeface conventions ............................................ xii
  Operating system differences .................................... xii

Chapter 1. pdadmin command line utility ....................... 1
pdadmin utility .................................................... 1
Command modes ..................................................... 1
  Single command mode .......................................... 1
  Interactive command mode ..................................... 2
  Multiple command mode ....................................... 2
Special characters disallowed for GSO commands ............... 3
Command option processing ...................................... 3
Error handling ..................................................... 3
  Return codes for a single command ............................. 4
  Return codes for an interactive command ....................... 4
  Return codes for multiple commands ............................. 4
Tivoli Access Manager pdadmin commands ...................... 5
  Access control list commands ................................ 5
  Action commands .............................................. 6
  Group commands ............................................... 6
  Login / logout commands ..................................... 6
  Object commands .............................................. 7
  Object space commands ...................................... 7
  Policy commands ............................................... 7
  Protected object policy commands .............................. 7
  Resource commands ........................................... 8
  Server commands .............................................. 8
  User commands ................................................. 9
acl attach ....................................................... 10
acl create ....................................................... 11
acl delete ....................................................... 12
acl detach ....................................................... 13
acl find ......................................................... 14
acl list ......................................................... 15
acl modify ....................................................... 16
acl show ......................................................... 21
action create .................................................... 23
action delete ..................................................... 25
action group ..................................................... 26

© Copyright IBM Corp. 2001, 2003
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>action list</td>
<td>27</td>
</tr>
<tr>
<td>admin show conf.</td>
<td>28</td>
</tr>
<tr>
<td>errtext.</td>
<td>29</td>
</tr>
<tr>
<td>exit / quit</td>
<td>30</td>
</tr>
<tr>
<td>group create</td>
<td>31</td>
</tr>
<tr>
<td>group delete</td>
<td>33</td>
</tr>
<tr>
<td>group import</td>
<td>34</td>
</tr>
<tr>
<td>group list</td>
<td>35</td>
</tr>
<tr>
<td>group modify</td>
<td>37</td>
</tr>
<tr>
<td>group show.</td>
<td>39</td>
</tr>
<tr>
<td>help</td>
<td>41</td>
</tr>
<tr>
<td>login</td>
<td>43</td>
</tr>
<tr>
<td>logout</td>
<td>44</td>
</tr>
<tr>
<td>object create</td>
<td>45</td>
</tr>
<tr>
<td>object delete</td>
<td>47</td>
</tr>
<tr>
<td>object list</td>
<td>48</td>
</tr>
<tr>
<td>object listandshow</td>
<td>50</td>
</tr>
<tr>
<td>object modify</td>
<td>52</td>
</tr>
<tr>
<td>object show</td>
<td>55</td>
</tr>
<tr>
<td>objectspace create</td>
<td>57</td>
</tr>
<tr>
<td>objectspace delete</td>
<td>59</td>
</tr>
<tr>
<td>objectspace list</td>
<td>60</td>
</tr>
<tr>
<td>policy get</td>
<td>61</td>
</tr>
<tr>
<td>policy set</td>
<td>63</td>
</tr>
<tr>
<td>pop attach</td>
<td>66</td>
</tr>
<tr>
<td>pop create</td>
<td>68</td>
</tr>
<tr>
<td>pop delete</td>
<td>69</td>
</tr>
<tr>
<td>pop detach</td>
<td>70</td>
</tr>
<tr>
<td>pop find</td>
<td>71</td>
</tr>
<tr>
<td>pop list</td>
<td>72</td>
</tr>
<tr>
<td>pop modify</td>
<td>73</td>
</tr>
<tr>
<td>pop show</td>
<td>76</td>
</tr>
<tr>
<td>rsrcc create</td>
<td>78</td>
</tr>
<tr>
<td>rsrcc delete</td>
<td>80</td>
</tr>
<tr>
<td>rsrcc list</td>
<td>81</td>
</tr>
<tr>
<td>rsrcc show</td>
<td>82</td>
</tr>
<tr>
<td>rsrcccred create</td>
<td>83</td>
</tr>
<tr>
<td>rsrcccred delete</td>
<td>85</td>
</tr>
<tr>
<td>rsrcccred list user</td>
<td>87</td>
</tr>
<tr>
<td>rsrcccred modify</td>
<td>88</td>
</tr>
<tr>
<td>rsrcccred show</td>
<td>90</td>
</tr>
<tr>
<td>rsrccgroup create</td>
<td>92</td>
</tr>
<tr>
<td>rsrccgroup delete</td>
<td>93</td>
</tr>
<tr>
<td>rsrccgroup list</td>
<td>94</td>
</tr>
<tr>
<td>rsrccgroup modify</td>
<td>95</td>
</tr>
<tr>
<td>rsrccgroup show</td>
<td>97</td>
</tr>
<tr>
<td>server list</td>
<td>98</td>
</tr>
<tr>
<td>server listtasks</td>
<td>99</td>
</tr>
<tr>
<td>server replicate</td>
<td>101</td>
</tr>
<tr>
<td>server show</td>
<td>102</td>
</tr>
<tr>
<td>server task</td>
<td>104</td>
</tr>
<tr>
<td>server task (WebSEAL)</td>
<td>105</td>
</tr>
<tr>
<td>server task add (WebSEAL)</td>
<td>107</td>
</tr>
<tr>
<td>server task stats (WebSEAL)</td>
<td>109</td>
</tr>
<tr>
<td>server task trace</td>
<td>111</td>
</tr>
<tr>
<td>user create</td>
<td>113</td>
</tr>
</tbody>
</table>
Preface

IBM® Tivoli® Access Manager (Tivoli Access Manager) is the base software that is required to run applications in the IBM Tivoli Access Manager product suite. It enables the integration of IBM Tivoli Access Manager applications that provide a wide range of authorization and management solutions. Sold as an integrated solution, these products provide an access control management solution that centralizes network and application security policy for e-business applications.

Note: IBM Tivoli Access Manager is the new name of the previously released software entitled Tivoli SecureWay® Policy Director. Also, for users familiar with the Tivoli SecureWay Policy Director software and documentation, the management server is now referred to as the policy server.

The IBM Tivoli Access Manager Command Reference provides detailed information about the pdadmin command line interface and other command line utilities, which can help you manage servers and resources in your secure domain.

Who should read this book

This reference is for system administrators responsible for the administration of Tivoli Access Manager software.

Readers should be familiar with the following:
- Microsoft® Windows® and UNIX® operating systems
- Database architecture and concepts
- Security management
- Internet protocols, including HTTP, HTTPS, TCP/IP, File Transfer Protocol (FTP), and Telnet
- Lightweight Directory Access Protocol (LDAP) and directory services
- Authentication and authorization
- Access Manager security model and its capabilities

If you are enabling Secure Sockets Layer (SSL) communication, you also should be familiar with SSL protocol, key exchange (public and private), digital signatures, cryptographic algorithms, and certificate authorities.

What this book contains

This reference contains the following sections:
- Chapter 1, “pdadmin command line utility”, on page 1
  Provides reference information about pdadmin commands.
- Chapter 2, “Tivoli Access Manager utilities”, on page 125
  Lists other Tivoli Access Manager utilities that can help you maintain your environment and troubleshoot problems that can arise during normal operations.
- “User registry differences” on page 163
  Documents the user registry differences that are known to exist in this version of Tivoli Access Manager.
Publications

The Tivoli Access Manager library is organized into the following categories:

- "Release information"
- "Base information"
- "WebSEAL information"
- "Web security information"
- "Developer references" on page ix
- "Technical supplements" on page ix

Release information

- IBM Tivoli Access Manager Read Me First Card
  GI11-4198-00 (am41_readme.pdf)
  Provides information for installing and getting started using Tivoli Access Manager.

- IBM Tivoli Access Manager Release Notes
  SC32-1130-00 (am41_relnotes.pdf)
  Provides late-breaking information, such as software limitations, workarounds, and documentation updates.

Base information

- IBM Tivoli Access Manager Base Installation Guide
  SC32-1131-01 (am41_install.pdf)
  Explains how to install, configure, and upgrade Tivoli Access Manager software, including the Web Portal Manager interface.

- IBM Tivoli Access Manager Base Administrator's Guide
  SC32-1132-01 (am41_admin.pdf)
  Describes the concepts and procedures for using Tivoli Access Manager services. Provides instructions for performing tasks from the Web Portal Manager interface and by using the pdadmin command.

WebSEAL information

- IBM Tivoli Access Manager WebSEAL Installation Guide
  SC32-1133-01 (amweb41_install.pdf)
  Provides installation, configuration, and removal instructions for the WebSEAL server and the WebSEAL application development kit.

- IBM Tivoli Access Manager WebSEAL Administrator's Guide
  SC32-1134-01 (amweb41_admin.pdf)
  Provides background material, administrative procedures, and technical reference information for using WebSEAL to manage the resources of your secure Web domain.

Web security information

- IBM Tivoli Access Manager for WebSphere Application Server User's Guide
  SC32-1136-01 (amwas41_user.pdf)
  Provides installation, removal, and administration instructions for Tivoli Access Manager for IBM WebSphere® Application Server.

- IBM Tivoli Access Manager for WebLogic Server User's Guide
  SC32-1137-01 (amwls41_user.pdf)
  Provides installation, removal, and administration instructions for Tivoli Access Manager for BEA WebLogic Server.
• **IBM Tivoli Access Manager Plug-in for Edge Server User's Guide**
  SC32-1138-01 (amedge41_user.pdf)
  Describes how to install, configure, and administer the plug-in for IBM WebSphere Edge Server application.

• **IBM Tivoli Access Manager Plug-in for Web Servers User's Guide**
  SC32-1139-01 (amws41_user.pdf)
  Provides installation instructions, administration procedures, and technical reference information for securing your Web domain using the plug-in for Web servers.

**Developer references**

• **IBM Tivoli Access Manager Authorization C API Developer's Reference**
  SC32-1140-01 (am41_authC_devref.pdf)
  Provides reference material that describes how to use the Tivoli Access Manager authorization C API and the Access Manager service plug-in interface to add Tivoli Access Manager security to applications.

• **IBM Tivoli Access Manager Authorization Java Classes Developer's Reference**
  SC32-1141-01 (am41_authJ_devref.pdf)
  Provides reference information for using the Java™ language implementation of the authorization API to enable an application to use Tivoli Access Manager security.

• **IBM Tivoli Access Manager Administration C API Developer's Reference**
  SC32-1142-01 (am41_adminC_devref.pdf)
  Provides reference information about using the administration API to enable an application to perform Tivoli Access Manager administration tasks. This document describes the C implementation of the administration API.

• **IBM Tivoli Access Manager Administration Java Classes Developer's Reference**
  SC32-1143-01 (am41_adminJ_devref.pdf)
  Provides reference information for using the Java language implementation of the administration API to enable an application to perform Tivoli Access Manager administration tasks.

• **IBM Tivoli Access Manager WebSEAL Developer's Reference**
  SC32-1135-01 (amweb41_devref.pdf)
  Provides administration and programming information for the Cross-domain Authentication Service (CDAS), the Cross-domain Mapping Framework (CDMF), and the Password Strength Module.

**Technical supplements**

• **IBM Tivoli Access Manager Command Reference**
  GC32-1107-01 (am41_cmdref.pdf)
  Provides information about the command line utilities and scripts provided with Tivoli Access Manager.

• **IBM Tivoli Access Manager Error Message Reference**
  SC32-1144-01 (am41_error_ref.pdf)
  Provides explanations and recommended actions for the messages produced by Tivoli Access Manager.

• **IBM Tivoli Access Manager Problem Determination Guide**
  GC32-1106-01 (am41_pdg.pdf)
  Provides problem determination information for Tivoli Access Manager.

• **IBM Tivoli Access Manager Performance Tuning Guide**
  SC32-1145-01 (am41_perftune.pdf)
  Provides performance tuning information for an environment consisting of Tivoli Access Manager with the IBM Directory server defined as the user registry.

**Related publications**
This section lists publications related to the Tivoli Access Manager library.
The Tivoli Software Library provides a variety of Tivoli publications such as white papers, datasheets, demonstrations, redbooks, and announcement letters. The Tivoli Software Library is available on the Web at: [http://www.ibm.com/software/tivoli/library/](http://www.ibm.com/software/tivoli/library/)

The Tivoli Software Glossary includes definitions for many of the technical terms related to Tivoli software. The Tivoli Software Glossary is available, in English only, from the Glossary link on the left side of the Tivoli Software Library Web page [http://www.ibm.com/software/tivoli/library/](http://www.ibm.com/software/tivoli/library/)

**IBM Global Security Toolkit**

Tivoli Access Manager provides data encryption through the use of the IBM Global Security Toolkit (GSKit). GSKit is included on the IBM Tivoli Access Manager Base CD for your particular platform.

The GSKit package installs the iKeyman key management utility, gsk5ikm, which enables you to create key databases, public-private key pairs, and certificate requests. The following document is available on the Tivoli Information Center Web site in the same section as the IBM Tivoli Access Manager product documentation:

- Secure Sockets Layer Introduction and iKeyman User’s Guide (gskikm5c.pdf)

  Provides information for network or system security administrators who plan to enable SSL communication in their Tivoli Access Manager environment.

**IBM DB2 Universal Database**

IBM DB2® Universal Database™ is required when installing IBM Directory Server, z/OS™, and OS/390® LDAP servers. DB2 is provided on the product CDs for the following operating system platforms:

- IBM AIX®
- Microsoft™ Windows™
- Sun Solaris Operating Environment

DB2 information is available at: [http://www.ibm.com/software/data/db2/](http://www.ibm.com/software/data/db2/)

**IBM Directory Server**


If you plan to use IBM Directory Server as your user registry, see the information provided at: [http://www.ibm.com/software/network/directory/library/](http://www.ibm.com/software/network/directory/library/)

**IBM WebSphere Application Server**


**IBM Tivoli Access Manager for Business Integration**

IBM Tivoli Access Manager for Business Integration, available as a separately orderable product, provides a security solution for IBM MQSeries®, Version 5.2, and IBM WebSphere® MQ for Version 5.3 messages. IBM Tivoli Access Manager for Business Integration allows WebSphere MQSeries applications to send data with privacy and integrity by using keys associated with sending and receiving applications. Like
WebSEAL and IBM Tivoli Access Manager for Operating Systems, IBM Tivoli Access Manager for Business Integration, is one of the resource managers that use the authorization services of IBM Tivoli Access Manager for e-business.

The following documents associated with IBM Tivoli Access Manager for Business Integration Version 4.1 are available on the Tivoli Information Center Web site:

- IBM Tivoli Access Manager for Business Integration Administrator’s Guide (SC23-4831-00)
- IBM Tivoli Access Manager for Business Integration Release Notes (GI11-0957-00)
- IBM Tivoli Access Manager for Business Integration Read Me First (GI11-0958-00)

IBM Tivoli Access Manager for Operating Systems

IBM Tivoli Access Manager for Operating Systems, available as a separately orderable product, provides a layer of authorization policy enforcement on UNIX systems in addition to that provided by the native operating system. IBM Tivoli Access Manager for Operating Systems, like WebSEAL and IBM Tivoli Access Manager for Business Integration, is one of the resource managers that use the authorization services of IBM Tivoli Access Manager for e-business.

The following documents associated with IBM Tivoli Access Manager for Operating Systems Version 4.1 are available on the Tivoli Information Center Web site:

- IBM Tivoli Access Manager for Operating Systems Installation Guide (SC23-4829-00)
- IBM Tivoli Access Manager for Operating Systems Administration Guide (SC23-4827-00)
- IBM Tivoli Access Manager for Operating Systems Problem Determination Guide (SC23-4828-00)
- IBM Tivoli Access Manager for Operating Systems Release Notes (GI11-0951-00)
- IBM Tivoli Access Manager for Operating Systems Read Me First (GI11-0949-00)

Accessing publications online

The publications for this product are available online in Portable Document Format (PDF) or Hypertext Markup Language (HTML) format, or both in the Tivoli Software Library:

http://www.ibm.com/software/tivoli/library

To locate product publications in the library, click the Product manuals link on the left side of the Library page. Then, locate and click the name of the product on the Tivoli Software Information Center page.

Product publications include release notes, installation guides, user’s guides, administrator’s guides, and developer’s references.

Note: To ensure proper printing of PDF publications, select the Fit to page check box in the Adobe Acrobat Print window (which is available when you click File → Print).

Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You also can use the keyboard instead of the mouse to operate all features of the graphical user interface.

Contacting software support

Before contacting IBM Tivoli Software support with a problem, refer to the IBM Tivoli Software support Web site at: http://www.ibm.com/software/sysmgmt/products/support

If you need additional help, contact software support by using the methods described in the IBM Software Support Guide at the following Web site: http://techsupport.services.ibm.com/guides/handbook.html
The guide provides the following information:

- Registration and eligibility requirements for receiving support
- Telephone numbers and e-mail addresses, depending on the country in which you are located
- A list of information you should gather before contacting customer support

Conventions used in this book

This reference uses several conventions for special terms and actions and for operating system-dependent commands and paths.

Typeface conventions

The following typeface conventions are used in this reference:

**Bold** Lowercase commands or mixed case commands that are difficult to distinguish from surrounding text, keywords, parameters, options, names of Java classes, and objects are in **bold**.

*Italic* Variables, titles of publications, and special words or phrases that are emphasized are in *italic*.

*Monospace* Code examples, command lines, screen output, file and directory names that are difficult to distinguish from surrounding text, system messages, text that the user must type, and values for arguments or command options are in monospace.

Operating system differences

This book uses the UNIX convention for specifying environment variables and for directory notation. When using the Windows command line, replace $variable with %variable% for environment variables and replace each forward slash (/) with a backslash (\) in directory paths. If you are using the bash shell on a Windows system, you can use the UNIX conventions.
Chapter 1. pdadmin command line utility

The pdadmin command line utility is installed as part of the IBM Tivoli Access Manager (Tivoli Access Manager) runtime package. Use this interface to manage access control lists, groups, servers, users, objects, and other resources in your secure domain. You can also automate certain management functions by writing scripts that use pdadmin commands.

Note that the Web Portal Manager interface, discussed in the IBM Tivoli Access Manager Base Administrator’s Guide enables you to perform similar administrative tasks remotely, without requiring any special network configuration.

pdadmin utility

USAGE:

```
pdadmin [–a admin_id [–p password] [–v] [cmd|file]

–a admin_id
Logs you in as the user admin_id. If you do not specify this option on the command line, you are considered unauthenticated, and your access to other commands is limited.

–p password
Specifies the password for the user admin_id. Using this option might expose your password to others because the password is visible on the screen and also in the process table. If you do not specify this option on the command line, you are prompted for a password. This option cannot be used unless the –a option is used.

–linelen max–linelen
This option is currently ignored.

–v
Prints out the version number of the pdadmin utility. If this option is specified, all other valid options are ignored.

The following example is output you might see when you use this option:

Access Manager Administrative Tool v4.1.0 (Build 020923)
Copyright (C) IBM Corporation 1994-2002. All Rights Reserved.

cmd | file
Specifies a single pdadmin command to execute (the cmd argument) or a file containing a list of commands to execute (the “file” argument). The command or commands are executed one time, and pdadmin does not enter interactive mode. Only one of the following can be specified: cmd or file.
```

Command modes

You can use the pdadmin command line interface in one of the following three modes:

• Single command mode
• Interactive command mode
• Multiple command mode

These modes are described in the following sections.

Single command mode

To run a single pdadmin command from a command prompt, type the following:

```
pdadmin [–a admin_id [–p password] [command]
```

Where:
-a admin_id  Logs you in as the user admin_id. If you do not specify this option on the command line, you are considered unauthenticated, and your access to other commands is limited.

-p password  Specifies the password for the user admin_id. Using this option might expose your password to others because the password is visible on the screen and also in the process table. If you do not specify this option on the command line, you are prompted for a password. This option cannot be used unless the –a option is used.

command  Allows you to run one-time commands. For example, user chris is created if you type the following command, all on one line.

    pdadmin –a sec_master –p password user create chris
    cn=chris,ou=austin,o=ibm,c=us
    chris chris chris1234

Notes:
- If you specify the admin_id (–a) and password (–p), you are logged in as that user. Using this method might expose your password to others. For example, if one user is using pdadmin with this command, and another user lists the processes that are running, the full command (including the password), might be visible to that user.
- If you do not specify the admin_id (–a), you are logged in as an unauthenticated user.
- If you specify the admin_id (–a), but do not specify a password (–p), you are prompted for a password.

Interactive command mode
To start pdadmin in interactive mode, type the pdadmin command followed by the login command as follows:

    pdadmin
    login –a admin_id –p <password>
    pdadmin>

See “Login / logout commands” on page 6 for additional information about the login command. At the pdadmin prompt, type the appropriate commands and their associated options.

Multiple command mode
You can create a file that contains multiple pdadmin commands, one per line, that together perform a complete task or series of tasks.

To run commands in this file, type the following:

    pdadmin [-a admin_id [-p password]] filename

Where:
- a admin_id  Logs you in as the user admin_id. If you do not specify this option on the command line, you are considered unauthenticated, and your access to other commands is limited.

-p password  Specifies the password for the user admin_id. Using this option might expose your password to others because the password is visible on the screen and also in the process table. If you do not specify this option on the command line, you are prompted for a password. This option cannot be used unless the –a option is used.

filename  Specifies the complete path and name of the file containing the pdadmin commands.

A valid file name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period
Notes:
• If you specify the `admin_id` (–a) and password (–p), you are logged in as that user.
• If you do not specify the `admin_id` (–a), you are logged in as an unauthenticated user.
• If you specify the `admin_id` (–a), but do not specify a password (–p), you are prompted for a password.

Special characters disallowed for GSO commands
You cannot use the following characters to create a global signon (GSO) user name, GSO resource name, or GSO resource group name:

`! " # & ( ) * , + , ; : < > = ? @ [ ] ^ _ { | `}

Although it is possible to use most of these characters for other LDAP-related data, such as the common name (CN), distinguished name (DN), and short name (SN) of a user, these characters have special meaning in LDAP DN syntax and filters.

Before using any of these characters in user and group names, consult the documentation for your user registry to determine the effect of special characters.

Command option processing
Some `pdadmin` command options begin with a hyphen (–). For example, the following command uses the –gsouser option:

```
pdadmin> user import –gsouser mlucaser cn=mlucaser,ou=Austin,o=Tivoli,c=US
```

The `pdadmin` command interprets any token beginning with a hyphen as a command option, even if the hyphen is placed within double quotation marks.

Occasionally, you might want a token that begins with a – to be interpreted as an argument rather than as a command option. For example, you might want to name the user `–mlucaser` or "–mlucaser" by typing:

```
pdadmin> user import –gsouser –mlucaser cn=mlucaser,ou=Austin,o=ibm,c=us
```

In this example, the first –gsouser option in the command is still processed. However, because the user name token begins with a hyphen, the user name would be interpreted as a command option. The command would fail because the –mlucaser command option does not exist.

You can specify the single hyphen character to turn off the `pdadmin` command’s interpretation of the optional arguments. Following the single hyphen character, –mlucaser is now interpreted as the user name. For example:

```
pdadmin> user import –gsouser – –mlucaser cn=mlucaser,ou=Austin,o=Tivoli,c=us
```

Options on the command line are position-independent. You can change the order so that all tokens that begin with a hyphen, which are not command options, follow the single hyphen character.

Error handling
The `pdadmin` command has two return code values:
0 The command completed successfully.

1 The command failed. When a command fails, the `pdadmin` command displays a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

Also see the `pdadmin errtext` command for an explanation of how you can use the message number that is associated with a message as input to display only the descriptive text.

### Return codes for a single command

A single command is normally typed from a DOS command prompt, Korn shell prompt, C shell prompt, and so forth. Single command mode does not automatically display the 0 or 1 return code values; the operating system must be queried for the return code value.

For command failures, only the hexadecimal error code status with its associated error message is shown. You can redirect the error that is normally displayed on the screen out to a text file. When a single command fails, you see an error message similar to the following displayed:

```plaintext
C:> pdadmin -a admin_id -p password user show oogle
Could not perform the administration request.
Error: Not found. (status 0x14c012f2)
```

To display the 0 or 1 return code values, you must type the `pdadmin` command, followed by either the UNIX `echo` or the Windows `errorlevel` command:

- For UNIX:
  ```
  # pdadmin_command
  # echo $?  
  ```

- For Windows:
  ```
  C:>pdadmin_command
  C:>echo %errorlevel%
  ```

### Return codes for an interactive command

Interactive command mode does not automatically display the 0 or 1 return code values. Neither can you follow an interactive command with the UNIX `echo` nor the Windows `errorlevel` command.

For a command failure, you see a message similar to:

```plaintext
pdadmin> user show oogle
Could not perform the administration request.
Error: Not found. (status 0x14c012f2)
```

Only the hexadecimal exit status code is displayed.

### Return codes for multiple commands

You can use a text file containing `pdadmin` commands to run those commands in a single `pdadmin` invocation. If an error occurs for any command while running the commands (multiple command mode), an error message for the failed command will be provided.

Processing of the remaining commands in the file continues after an error. At the end of multiple command processing, a final status is provided. Note that the final status code at the termination of multiple command processing is only for the last command that was attempted. For example, if the last command was successful, the final status will be 0; if the last command failed, the final status will be 1.

For example, a text file might contain the following `pdadmin` commands:

```plaintext
user show cwright
user show oogle
```
To run the commands, run the following command:

```
padmin -a admin_id -p password cmd_filename
```

The above command would produce the results similar to the following:

The script file would produce the results similar to the following:

```
cmd> user show cwright
Login ID: cwright
LDAP DN: cn=Claude Wright,ou=Dallas,o=Tivoli,c=us
LDAP CN: Claude Wright
LDAP SN: Wright
Description:
  Is SecUser: yes
  Is GSO user: no
  Account valid: yes
  Password valid: yes
  Authorization mechanism: Default:LDAP

cmd:> user show oogle
Could not perform the administration request.
Error: Not found. (status 0x14c012f2)
```

### Tivoli Access Manager pdadmin commands

This section lists Tivoli Access Manager `pdadmin` commands by category and by command name.

- **"Access control list commands" on page 5**
- **"Action commands" on page 6**
- **"Group commands" on page 6**
- **"Login / logout commands" on page 6**
- **"Object commands" on page 7**
- **"Object space commands" on page 7**
- **"Policy commands" on page 7**
- **"Protected object policy commands" on page 7**
- **"Resource commands" on page 8**
- **"Server commands" on page 8**
- **"User commands" on page 9**

#### Access control list commands

Table 1 lists `acl` commands, which enable you to manage access control list (ACL) policies and extended attributes.

Table 1. Access control list (ACL) commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl attach</td>
<td>Attaches an ACL policy to a protected object. If the protected object already has an ACL attached, the ACL is replaced with a new one.</td>
</tr>
<tr>
<td>acl create</td>
<td>Creates an ACL policy in the ACL database. This command does not create ACL entries.</td>
</tr>
<tr>
<td>acl delete</td>
<td>Deletes an ACL policy from the ACL database.</td>
</tr>
<tr>
<td>acl detach</td>
<td>Detaches the current ACL policy from a protected object. This command does not delete the ACL policy from the ACL database.</td>
</tr>
<tr>
<td>acl find</td>
<td>Finds and lists all protected objects that have a specific ACL policy attached.</td>
</tr>
</tbody>
</table>
Table 1. Access control list (ACL) commands (continued)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl list</td>
<td>Lists the names of all defined ACLs. Also lists the extended attribute keys associated with a specific ACL.</td>
</tr>
<tr>
<td>acl modify</td>
<td>Modifies ACLs, their extended attributes, and associated values.</td>
</tr>
<tr>
<td>acl show</td>
<td>Lists the complete set of entries for a specific ACL policy. Also lists the values of a specific extended attribute associated with an ACL policy.</td>
</tr>
</tbody>
</table>

**Action commands**

Table 2 lists **action** commands, which are used to define additional authorization actions (ACL permissions) and action groups.

Table 2. Action commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action create</td>
<td>Defines an action (permission) code in an action group. Also adds an action code to a specific extended action group.</td>
</tr>
<tr>
<td>action delete</td>
<td>Deletes an action code for an action group. Also defines a specific action group from which to delete an action.</td>
</tr>
<tr>
<td>action group</td>
<td>Creates, deletes, and lists ACL action groups.</td>
</tr>
<tr>
<td>action list</td>
<td>Lists all defined action codes for an action group.</td>
</tr>
</tbody>
</table>

**Group commands**

A **group** is a set of Tivoli Access Manager user accounts that have similar attributes. Groups allow you to use a group name in an access control list (ACL) instead of listing all users individually. When an LDAP-based user registry is used, group names are not case sensitive.

Table 3 lists **group** commands, which are used to manage Tivoli Access Manager groups.

Table 3. Group commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>group create</td>
<td>Creates a Tivoli Access Manager group.</td>
</tr>
<tr>
<td>group delete</td>
<td>Deletes the specified Tivoli Access Manager group and optionally deletes the information about the group from the user registry. ACL entries associated with the group are also deleted.</td>
</tr>
<tr>
<td>group import</td>
<td>Imports the information about an existing registry group to create a Tivoli Access Manager group.</td>
</tr>
<tr>
<td>group list</td>
<td>Generates a list of all groups, by group names, whose names match the specified pattern.</td>
</tr>
<tr>
<td>group modify</td>
<td>Changes an existing group by adding a description, or adding or removing a list of members.</td>
</tr>
<tr>
<td>group show</td>
<td>Displays details about a specified group.</td>
</tr>
</tbody>
</table>

**Login / logout commands**

Table 4 on page 7 lists **login** and **logout** commands, which are used to login to and log out of a Tivoli Access Manager secure domain.
Table 4. Login commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>login</td>
<td>Authenticates the user to the Tivoli Access Manager policy server as a given administrative identity in a given domain.</td>
</tr>
<tr>
<td>logout</td>
<td>Discards any authentication credentials that are in effect.</td>
</tr>
</tbody>
</table>

Object commands

Table 5 lists objects commands, which enable you to protect objects by attaching ACLs or protected object policy (POP).

Table 5. Object commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object create</td>
<td>Creates a protected object.</td>
</tr>
<tr>
<td>object delete</td>
<td>Deletes a protected object.</td>
</tr>
<tr>
<td>object list</td>
<td>Lists any objects grouped under the specified protected object. Also lists all the extended attributes associated with the specified protected object.</td>
</tr>
<tr>
<td>object listandshow</td>
<td>Lists any child objects grouped under the specified protected object and displays all values associated with each of those objects.</td>
</tr>
<tr>
<td>object modify</td>
<td>Modifies an existing object.</td>
</tr>
<tr>
<td>object show</td>
<td>Shows all values associated with a protected object.</td>
</tr>
</tbody>
</table>

Object space commands

Table 6 lists objectspace commands, which allow the creation of additional object spaces containing protected objects used by third-party applications.

Table 6. Objectspace commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>objectspace create</td>
<td>Creates a protected object space under which protected objects can be placed.</td>
</tr>
<tr>
<td>objectspace delete</td>
<td>Deletes an existing protected object space and all associated protected objects.</td>
</tr>
<tr>
<td>objectspace list</td>
<td>Lists all of the existing protected object spaces in the policy server.</td>
</tr>
</tbody>
</table>

Policy commands

Table 7 lists policy commands, which are used to manage user password and account policies.

Table 7. Policy commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>policy get</td>
<td>Displays user password, account rules, and conditions.</td>
</tr>
<tr>
<td>policy set</td>
<td>Sets user password, account rules, and conditions.</td>
</tr>
</tbody>
</table>

Protected object policy commands

Table 8 lists pop commands, which allow the creation of a protected object policy (POP) and extended attributes for the protected object policies.

Table 8. Protected object policy (POP) commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pop attach</td>
<td>Attaches a protected object policy to a specified protected object.</td>
</tr>
</tbody>
</table>
Table 8. Protected object policy (POP) commands (continued)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pop create</code></td>
<td>Creates a protected object policy.</td>
</tr>
<tr>
<td><code>pop delete</code></td>
<td>Deletes the specified protected object policy.</td>
</tr>
<tr>
<td><code>pop detach</code></td>
<td>Detaches a protected object policy from the specified protected object.</td>
</tr>
<tr>
<td><code>pop find</code></td>
<td>Finds and lists all protected objects that have protected object policies attached.</td>
</tr>
<tr>
<td><code>pop list</code></td>
<td>Lists all protected object policies that have been created.</td>
</tr>
<tr>
<td><code>pop modify</code></td>
<td>Modifies the protected object policy.</td>
</tr>
<tr>
<td><code>pop show</code></td>
<td>Shows details of the protected object policy.</td>
</tr>
</tbody>
</table>

**Resource commands**

Table 9 lists resource commands, which are used to manage resource-related information.

Table 9. Resource commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>rsrc create</code></td>
<td>Creates and names a server as a resource.</td>
</tr>
<tr>
<td><code>rsrc delete</code></td>
<td>Deletes the specified single signon resource.</td>
</tr>
<tr>
<td><code>rsrc list</code></td>
<td>Returns a list of all the single signon resource names.</td>
</tr>
<tr>
<td><code>rsrc show</code></td>
<td>Displays the resource information for the named resource.</td>
</tr>
<tr>
<td><code>rsrcgroup create</code></td>
<td>Creates and names a resource group.</td>
</tr>
<tr>
<td><code>rsrcgroup delete</code></td>
<td>Deletes the named resource group, including any description information.</td>
</tr>
<tr>
<td><code>rsrcgroup modify</code></td>
<td>Adds or removes a single signon resource to or from a single signon resource group.</td>
</tr>
<tr>
<td><code>rsrcgroup list</code></td>
<td>Displays the names of all resource groups defined in the user registry.</td>
</tr>
<tr>
<td><code>rsrcgroup show</code></td>
<td>Displays the resource group information for the specified resource group.</td>
</tr>
<tr>
<td><code>rsrccred create</code></td>
<td>Creates and names a resource credential.</td>
</tr>
<tr>
<td><code>rsrccred modify</code></td>
<td>Changes the user ID and password resource credential information for the named resource.</td>
</tr>
<tr>
<td><code>rsrccred delete</code></td>
<td>Deletes only the resource credential information for an existing user.</td>
</tr>
<tr>
<td><code>rsrccred list user</code></td>
<td>Displays the names of all defined resources and their type for the specified user.</td>
</tr>
<tr>
<td><code>rsrccred show</code></td>
<td>Displays the resource credential information for a specified user.</td>
</tr>
</tbody>
</table>

**Server commands**

Table 10 lists server commands and the `admin show configuration` command, which perform management tasks on Tivoli Access Manager servers.

Table 10. Server commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>admin show conf</code></td>
<td>Displays current policy server configuration information.</td>
</tr>
<tr>
<td><code>server list</code></td>
<td>Lists all registered servers.</td>
</tr>
<tr>
<td><code>server list</code></td>
<td>Retrieves the list of tasks (commands) available for this server.</td>
</tr>
<tr>
<td><code>server replicate</code></td>
<td>Notifies authorization servers to receive database updates.</td>
</tr>
<tr>
<td><code>server show</code></td>
<td>Displays the specified properties of the server.</td>
</tr>
</tbody>
</table>
Table 10. Server commands (continued)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>server task</td>
<td>Sends the specified command to the specified server. WebSEAL-specific options are also included.</td>
</tr>
</tbody>
</table>

**User commands**

Table 11 lists **user** commands, which are used to manage Tivoli Access Manager users.

Table 11. User commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user create</td>
<td>Creates a Tivoli Access Manager user account.</td>
</tr>
<tr>
<td>user delete</td>
<td>Deletes a Tivoli Access Manager user and optionally deletes the user information from the user registry. ACL entries associated with the user are also deleted.</td>
</tr>
<tr>
<td>user import</td>
<td>Imports the information about an existing registry user to create a Tivoli Access Manager user.</td>
</tr>
<tr>
<td>user list</td>
<td>Generates a list of all users whose names match the specified pattern, listed by user names.</td>
</tr>
<tr>
<td>user modify</td>
<td>Modifies various user account parameters.</td>
</tr>
<tr>
<td>user show</td>
<td>Displays details about a specified user.</td>
</tr>
</tbody>
</table>
acl attach

Attaches an ACL policy to a protected object. If the protected object already has an ACL attached, the ACL is replaced with a new one.

Syntax

```
pdadmin -a admin_id -p password acl attach object_name acl_name
```

Options

- **-a admin_id** Specifies the administrator's ID. If this is the only option that is specified, the user is prompted for the password.
- **-p password** Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.
- **object_name** Specifies the object to which to apply the named ACL policy.

A valid object name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of object names: /Management/Groups/Travel, /WebSEAL, and /Management

- **acl_name** Specifies the ACL policy that is applied to the named object.

A valid ACL policy name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples: default-root, test, default-management, pubs_acl3

Description

At most, one ACL can be attached to a given protected object. The same ACL can be attached to multiple protected objects. Ensure that you are familiar with ACL management before using this function.

Examples

The following example attaches the ACL policy pubs_acl3 to the protected object /Management.

```
pdadmin> acl attach /Management pubs_acl3
```

Return Codes

The following exit status codes can be returned:

- **0** The command completed successfully.
- **1** The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

- acl create
- acl detach
acl create

Creates an ACL policy in the ACL database. Note that this command does not create ACL entries.

Syntax

```
pdadmin –a admin_id –p password acl create acl_name
```

Options

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `acl_name` Specifies the name of the ACL policy being created.

A valid ACL policy name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples:

1. The following example creates a new ACL policy named `pubs_acl3`:
   ```
pdadmin>acl create pubs_acl3
   ```
2. The following example creates a new ACL policy named `Test-ACL`:
   ```
pdadmin>acl create Test-ACL
   ```

Return Codes

The following exit status codes can be returned:

- `0` The command completed successfully.
- `1` The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

- acl attach
- acl delete
- acl modify
acl delete

Deletes an ACL policy from the ACL database.

Syntax

pdadmin –a admin_id –p password acl delete acl_name

Options

- a admin_id Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- p password Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.
acl_name Specifies the name of the ACL policy being deleted from the ACL database.

A valid ACL policy name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples: default-root, test, default-management, pubs_acl3

Examples

1. The following example deletes the ACL policy named pubs_acl3:
   pdadmin>acl delete pubs_acl3
2. The following example deletes the ACL policy named Test-ACL
   pdadmin>acl delete Test-ACL

Return Codes

The following exit status codes can be returned:

0 The command completed successfully.
1 The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference] This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

acl detach
acl detach

Detaches the current ACL policy from a protected object. Note that this command does not delete the ACL policy from the ACL database.

Syntax

```
pdadmin –a admin_id –p password acl detach object_name
```

Options

- `-a admin_id`  Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `-p password`  Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `object_name`  Specifies the object from which the current ACL policy is being removed. The ACL policy you want to detach must have previously been attached to the specified object.

A valid object name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of object names: /Management/Groups/Travel, /WebSEAL, and /Management

Description

Only one access control list at a time can be attached to an object. Therefore, the currently attached access control list is detached.

Examples

The following example detaches the protected object /Management.

```
pdadmin> acl detach /Management
```

Return Codes

The following exit status codes can be returned:

- **0**  The command completed successfully.
- **1**  The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

- acl attach
- acl delete
- acl modify
acl find

Returns a list of protected objects, which have the specified ACL attached.

Syntax

```
padmin –a admin_id –p password acl find acl_name
```

Options

- `–a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `–p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `acl_name` Specifies the name of the ACL policy that you want to find.

A valid ACL policy name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples: default-root, test, default-management, pubs_acl3

Examples

1. The following example lists the protected object that has the default-config ACL attached:

```
padmin> acl find default-config
/Management/Config
```

2. The following example lists the protected objects that have a user-defined ACL attached:

```
padmin> acl find _WebAppServer_deployedResources_CosNamingDelete_admin_ACL
/WebAppServer/deployedResources/CosNamingDelete/admin
```

Return Codes

The following exit status codes can be returned:

0 The command completed successfully.
1 The command failed. When a command fails, the `padmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

- `acl list`
- `acl show`
acl list

Lists the names of all defined access control lists. Alternatively, lists the extended attribute keys associated with a specific ACL.

Syntax

pdadmin –a admin_id –p password acl list [acl_name attribute]

Options

– a admin_id  Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

– p password  Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.

acl_name  Specifies the ACL policy for which to list the attributes.

A valid ACL policy name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples: default-root, test, default-management, pubs_acl3

Examples

The following example lists ACL policies:

pdadmin>acl list

default-webseal
default-root
test
default-replica
default-management

Return Codes

The following exit status codes can be returned:

0  The command completed successfully.

1  The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

acl find

acl show
acl modify
Modifies access control list (ACL) policies.

Syntax
pdadmin –a admin_id –p password acl modify acl_name delete attribute attribute_name [attribute_value]

pdadmin –a admin_id –p password acl modify acl_name description description

pdadmin –a admin_id –p password acl modify acl_name remove any-other

pdadmin –a admin_id –p password acl modify acl_name remove group group_name

pdadmin –a admin_id –p password acl modify acl_name remove unauthenticated

pdadmin –a admin_id –p password acl modify acl_name remove user user_name

pdadmin –a admin_id –p password acl modify acl_name set any-other [permissions]

pdadmin –a admin_id –p password acl modify acl_name set attribute attribute_name attribute_value

pdadmin –a admin_id –p password acl modify acl_name set description description

pdadmin –a admin_id –p password acl modify acl_name set group group_name [permissions]

pdadmin –a admin_id –p password acl modify acl_name set unauthenticated [permissions]

pdadmin –a admin_id –p password acl modify acl_name set user user_name [permissions]

Options

– a admin_id  Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

– p password  Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.

acl_name  Specifies the ACL policy that you want to be modified.

A valid ACL policy name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples: default-root, test, default-management, pubs_acl3

dele te attribute attribute_name [attribute_value]  Deletes the specified extended attribute name and value from the specified ACL.

A valid extended attribute name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

The optional attribute_value deletes the specified value from the specified extended attribute key in the specified ACL. A valid extended attribute value is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the
numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of extended attribute names and values:
Dept_No 445
Employee_Name "Diana Lucas"

description

Sets or modifies the description for the specified ACL. This option is equivalent to the acl modify set description command.

A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*). Spaces are allowed.

If the description contains a space, ensure that you enclose the description in double quotation marks. You can specify an empty string (""") to clear an existing description.

Example of description: "Department number of employee"

remove any-other

Removes the ACL entry for the user any-other from the specified ACL.

remove group group_name

Removes the ACL entry for the specified group from the specified ACL.

A valid group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of group names: Credit, Sales, Test-group

remove unauthenticated

Removes the ACL entry for the user unauthenticated from the specified ACL.

remove user user_name

Removes the ACL entry for the specified user from the specified ACL.

A valid user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of user names: dluca, sec_master, "Mary Jones"

set any-other [permissions]

Sets or modifies the ACL entry for the user any-other in the ACL.

Tivoli Access Manager uses a default set of actions that cover a wide range of operations. Valid actions, or permissions, are represented by single alphabetic ASCII characters (a-z, A-Z).

For example, a list primary action tasks and associated permissions for the user sec_master, with WebSEAL as the Web server, might include:

T Traverse Base
c Control Base
g Delegation Base
m Modify Generic
d Delete Generic
b Browse Base
s Server Admin Generic
v View Generic

Chapter 1. pdadmin command line utility 17
set attribute attribute_name attribute_value
Sets the extended attribute value for the specified extended attribute key in the specified ACL. If the attribute already exists, the attribute value is added as an additional value if the same value does not exist for this attribute. If the same value exists for this attribute, it does not get added again (duplicate values are not allowed), and no error is returned.

A valid extended attribute name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

The optional attribute_value deletes the specified value from the specified extended attribute key in the specified ACL. A valid extended attribute value is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of extended attribute names and values:
 Dept_No 445
 Employee_name "Diana Lucas"

set description description
Sets or modifies the description for the specified ACL.

A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*). Spaces are allowed.

If the description contains a space, ensure that you enclose the description in double quotation marks. You can specify an empty string ("") to clear an existing description.

Example of description: "Department number of employee"

set group group_name [permissions]
Sets or modifies the ACL entry for the specified group in the specified ACL. The user registry must contain an entry for the specified group before you can call this function to add an entry for the group to an ACL.

A valid group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of group names: Credit, Sales, Test-group

Tivoli Access Manager uses a default set of actions that cover a wide range of operations. Valid actions, or permissions, are represented by single alphabetic ASCII characters (a-z, A-Z). See the option set any-other[permissions] for examples of permissions.

set unauthenticated [permissions]
Sets or modifies the ACL entry for the user unauthenticated in the specified ACL.
Tivoli Access Manager uses a default set of actions that cover a wide range of operations. Valid actions, or permissions, are represented by single alphabetic ASCII characters (a-z, A-Z). See the option set any-other[permissions] for examples of permissions.

```
set user user_name [permissions]
```

Sets permissions that the user is permitted to perform. The user registry must contain an entry for the specified user before you can use this function to add an entry for the user to an ACL.

A valid user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of user names: dlucas, sec_master, "Mary Jones"

Tivoli Access Manager uses a default set of actions that cover a wide range of operations. Valid actions, or permissions, are represented by single alphabetic ASCII characters (a-z, A-Z). See the option set any-other[permissions] for examples of permissions.

Examples

1. The following example sets the any-other type for the ACL entry in the indicated ACL named pubs policy definition and sets permissions of r:

   pdadmin>acl modify pubs set any-other r

2. The following example sets a group type for the ACL entry in the indicated ACL named pubs policy definition and sets permissions of Tr:

   pdadmin>acl modify pubs set group sales Tr

3. The following example sets the unauthenticated type for the ACL entry in the indicated ACL named docs policy definition and sets permissions of r:

   pdadmin>acl modify docs set unauthenticated r

4. The following example sets a user type for the ACL entry in the indicated ACL named pubs policy definition and sets permissions of Tr:

   pdadmin>acl modify pubs set user peter Tr

5. The following example adds a ACL entry for user Kate containing actions from action groups primary and test group:

   pdadmin>acl modify and set user kathy brT\[ test-group\] PS
   pdadmin>acl show test
   ACL Name: test
   Description:
   Entries:
   User sec_master Tcmdbva
   Group ivmgrd-servers Tl
   Any-other r
   User kathy Tbr\[test-group\]PS

Return Codes

The following exit status codes can be returned:

0  The command completed successfully.

1  The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.
See Also

acl attach
acl create
acl show

Lists the complete set of entries for a specific access control list (ACL) policy. Alternatively, lists the values of a specific extended attribute associated with an ACL policy.

Syntax

pdadmin –a admin_id –p password acl show acl_name [attribute attribute_name]

Options

– a admin_id
  Specifies the administrator's ID. If this is the only option that is specified, the user is prompted for the password.

– p password
  Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.

acl_name
  Specifies the access control list for which the extended attribute values are displayed.

  A valid ACL policy name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples: default-root, test, default-management, pubs_acl3

attribute attribute_name
  Specifies the name of the extended attribute whose values you want to be displayed.

  A valid attribute name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of extended attribute names: Dept_No, Employee_Name

Examples

The following example shows details of ACL test:

pdadmin>acl show test
  ACL Name: test
  Description:
  Entries:
    User sec_master Tcmdbva
    Group ivmgrd-servers Tl
    Any other r

Return Codes

The following exit status codes can be returned:

  0  The command completed successfully.

  1  The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

  Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.
See Also

- acl find
- acl list
**action create**

Defines an action (permission) code in an action group. If you create an action without specifying an action group name, and then do an “action list primary”, the new action is in the list.

**Syntax**

```
pdadmin -a admin_id -p password action create action_name action_label action_type
[action_group_name]
```

**Options**

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- `action_name` Specifies the new single-character permission being created, which can be specified using any case.

Tivoli Access Manager uses a default set of actions that cover a wide range of operations. Valid actions, or permissions, are represented by single alphabetic ASCII characters (a-z, A-Z).

For example, `k` is the action name in the following example:

```
k time Ext-Authzn
```

- `action_label` Specifies the label or description for the action. Each default permission is displayed with a label describing the operation it governs. In addition, the ACLs are grouped according to their use in a particular part of the object space (such as WebSEAL) or their use across the entire object space (Base, Generic).

For example, `time` is the action label in the following example:

```
k time Ext-Authzn
```

A valid action label is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of action labels: time, Generic, Base, WebSEAL

- `action_type` Specifies the organizational category for this action within a given action group. The action type can be a description of the action, such as what application the action is specific to. The action type is application-specific and typically refers to the application that defined the action (such as WebSEAL) or the function that uses the action (such as Ext-Authzn for extended authorization checks).

A valid action type is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

For example, Ext-Authzn is the action type in the following example:

```
k time Ext-Authzn
```
Specifies the action group to which the action code is to be added. If no action group name is specified, the action will be added to the primary action group. Supports a maximum of 32 action groups.

A valid action group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of action group names: primary, test-group

Description
Action codes consist of one alphabetic character (a–z or A–Z) and are case-sensitive. Each action code only can be used once within an action group. Ensure that you do not attempt to redefine the default action codes when adding new codes to the primary group.

Examples
1. The following example creates a customized action code k with a label of time and puts the action with an action type of Ext-Authzn:
   pdadmin>action create k time Ext-Authzn
2. The following example creates an customized action named P with an action label of Test-Action with an action type of Special within an action group named test-group:
   pdadmin>action create P Test-Action Special test-group

Return Codes
The following exit status codes can be returned:
0  The command completed successfully.
1  The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x1c012f2).

Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
action delete
### action delete

Deletes an action (permission) code from an action group.

**Syntax**

```
padmin -a admin_id -p password action delete action_name [action_group_name]
```

**Options**

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `action_name` Specifies the name of the action to be deleted.
- `action_group_name` Specifies the name of the action group from which the specified action needs to be deleted.

Tivoli Access Manager uses a default set of actions that cover a wide range of operations. Valid actions, or permissions, are represented by single alphabetic ASCII characters (a-z, A-Z).

For example, `k` is the action name in the following example:

```
k time Ext-Authzn
```

Examples of action group names: primary, test-group

**Examples**

1. The following example deletes action `k` from the primary action group:
   ```
   pdadmin> action delete k
   ```

2. The following example deletes the action `z` from the action group `agz`:
   ```
   pdadmin> action delete z agz
   ```

**Return Codes**

The following exit status codes can be returned:

- **0** The command completed successfully.
- **1** The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the `IBM Tivoli Access Manager Error Message Reference`. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**

- `action create`
action group

Creates, deletes, and lists ACL action groups.

Syntax

```
pdadmin -a admin_id -p password action group { create action_group_name | delete action_group_name | list}
```

Options

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

`create action_group_name` Specifies the name of the action group to create. Supports a maximum of 32 action groups.

A valid action group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of action group names: `primary`, `test-group`

`delete action_group_name` Specifies the name of the action group to delete. All of the actions that belong to the specified group are also deleted.

A valid action group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of action group names: `primary`, `test-group`

`list` Lists all the defined action group names.

Examples

1. The following example lists the names of all defined action groups:

```
padmin>
padmin>action group list
primary
test-group
```

2. The following example creates an action group `test`:

```
padmin>action group create test
```

Return Codes

The following exit status codes can be returned:

0 The command completed successfully.

1 The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`).

Refer to the `IBM Tivoli Access Manager Error Message Reference` This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.
action list

Lists all the defined action (permission) codes from an action group.

Syntax

```plaintext
pdadmin -a admin_id -p password action list [action_group_name]
```

Options

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- `action_group_name` Specifies the name of the action group for which all actions are displayed. If this option is not specified, actions defined in the primary action group will be listed.

A valid action group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of action group names: `primary`, `test-group`

Examples

The following example displays all existing actions in the primary action group:

```plaintext
pdadmin>action list
T Traverse Base
c Control Base
g Delegation Base
m Modify Generic
d Delete Generic
b Browse Base
s Server Admin Generic
v View Generic
a Attach Base
B Bypass POP Base
t Trace Base
r Read WebSEAL
x Execute WebSEAL
l List Directory WebSEAL
N Create Base
W Password Base
A Add Base
```

Return Codes

The following exit status codes can be returned:

- **0** The command completed successfully.
- **1** The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#) This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.
admin show conf

Displays the current policy server configuration information, such as the type of registry or whether global signon is enabled.

Syntax

pdadmin –a admin_id –p password admin show conf

Options

–a admin_id Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

–p password Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.

Examples

The following example displays the current server configuration information:

pdadmin>admin show conf
LDAP: yes
secAuthority
GSO: yes

Return Codes

The following exit status codes can be returned:

0 The command completed successfully.

1 The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.
**errtext**

Displays the error message of a given error number. For detailed information on messages, see the IBM Tivoli Access Manager Error Message Reference.

### Syntax

`errtext error_number`

### Options

`error_number` Specifies the number, in either decimal or hexadecimal, of the error for which to generate the error text.

### Examples

1. The following example displays the error message associated with a given hexadecimal number:
   ```
   pdadmin>errtext 0x132120c8
   ```
   Output is similar to the following:
   ```
   Login failed. You have used an invalid user name, password or client certificate.
   ```

2. The following example displays the error message associated with a given decimal number:
   ```
   pdadmin>errtext 268809121
   ```
   Output is similar to the following:
   ```
   An attribute list already exists for this credential
   ```

### Return Codes

The following exit status codes can be returned:

- **0** The command completed successfully.
- **1** The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.
**exit / quit**

Exits from the **pdadmin** utility interactive command line mode.

**Syntax**

```
pdadmin {exit | quit}
```

**Options**

None.

**Examples**

1. The following example displays how to exit the **pdadmin** utility:
   
   ```
   pdadmin> exit
   ```

2. The following example displays how to quit the **pdadmin** utility:
   
   ```
   pdadmin> quit
   ```

**See Also**

- [login](#)
- [logout](#)
**group create**

Creates a Tivoli Access Manager group.

**Syntax**

```
padmin -a admin_id -p password group create group_name dn cn [group_container]
```

**Options**

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- `group_name` Specifies the name of the group being created. This name must be unique.

  A valid group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of group names: Credit, Sales, Test-group

- `dn` Specifies the registry identifier assigned to the group being created.

  The format for a distinguished name is similar to:

  "cn=credit,ou=Austin,o=Tivoli,c=US"

- `cn` Specifies the common name assigned to the group being created. For example: cwright

- `group_container_object` Specifies the group container object assigned to the group being created. If this option is not specified, the group by default is placed in the object space under /Management/Groups.

  Examples of group containers: Credit, Sales_Teams

**Examples**

1. The following example creates a group named `credit` with a common name of `credit` within an optional group container object named Credit:

   `padmin>group create credit "cn=credit,ou=Austin,o=Tivoli,c=US" Credit`

2. The following example creates a group named `sales` with a common name of `sales` within an optional group container object named `Sales_Teams`:

   `padmin>group create sales "cn=sales,o=tivoli,c=us" Sales_Teams`

**Return Codes**

The following exit status codes can be returned:

- **0** The command completed successfully.

- **1** The command failed. When a command fails, the `padmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

  Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.
See Also

- group delete
- group import
**group delete**

Deletes the specified Tivoli Access Manager group and optionally deletes the group’s information from the user registry. ACL entries associated with the group are also deleted.

**Syntax**

```
padmin -a admin_id -p password group delete [-registry] group_name
```

**Options**

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `-registry` Deletes the entire group object from the user registry.
- `group_name` Specifies the name of the Tivoli Access Manager group to be deleted.

A valid group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

**Examples**

1. The following example deletes the existing engineering group:

   ```
padmin>group delete engineering
   ```

2. The following example deletes the entire group object from the user registry and deletes the existing Test-group group:

   ```
padmin>action group delete -registry Test-group
   ```

**Return Codes**

The following exit status codes can be returned:

- `0` The command completed successfully.
- `1` The command failed. When a command fails, the `padmin` command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`).

Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**

- `group create`
- `group import`
**group import**

Creates a Tivoli Access Manager group by importing group data that already exists in the user registry.

**Syntax**

```bash
pdadmin -a admin_id -p password group import group_name dn [group_container]
```

**Options**

- `-a admin_id` Specifies the administrator's ID. If this is the only option that is specified, the user is prompted for the password.
- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `group_name` Specifies the Tivoli Access Manager name of the group to create.
  
  A valid group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of group names: Credit, Sales, Test-group

- `dn` Specifies the registry identifier of the group to import.

  The format for a distinguished name is similar to: "cn=Claude Wright,ou=Austin,o=Tivoli,c=us"

- `group_container` Specifies the group container object assigned to the group being created.

  By default, the group is placed in the object space under `/Management/Groups`. If the container object does not currently exist, it is automatically created.

**Examples**

1. The following example creates a Tivoli Access Manager group by importing a group that already exists in the user registry:

   ```bash
   pdadmin> group import engineering "cn=engineering,ou=Austin,o=Tivoli,c=us"
   ```

2. The following example creates a new group named `sales` and optionally places this group in a group container object named `Sales2002`.

   ```bash
   pdadmin> group import sales "cn=sales,o=tivoli,c=us" Sales2002
   ```

**Return Codes**

The following exit status codes can be returned:

- **0** The command completed successfully.
- **1** The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

  Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**

- [group create](#)
group list
Generates a list of all groups, by group names, whose names match the specified pattern.

Syntax
pdadmin –a admin_id –p password group {list | list-dn} pattern max_return

Options
–a admin_id  Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
–p password  Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.
list pattern max_return  Specifies the pattern for the group name for which to be searched. The pattern can include a mixture of wildcard and string constants, and is case insensitive (for example, *austin*).

The max_return option specifies the limit of how many entries should be returned for a single request; for example, 2. Note that the number returned is also governed by the server configuration, which specifies the maximum number of results that can be returned as part of a search operation. The actual maximum returned entries is the minimum of max_return and the configured value on the server.

list-dn pattern max_return  Lists user registry identifiers whose user registry common name attribute matches the pattern specified. The returned list are groups, which are defined in the user registry but are not necessarily Tivoli Access Manager groups. Groups that are not Tivoli Access Manager groups can be imported into Tivoli Access Manager by use of the group import command.

Examples
1. The following example lists 3 groups matching the specified pattern of a group name containing the letter a:
   pdadmin>group list *a* 3

   Output is similar to the following:
   sales
   marketing
   Alex
2. The following example lists 2 groups matching the specified pattern of a distinguished name containing the letter t:
   pdadmin>group list-dn *t* 2

   Output is similar to the following:
   cn=credit,ou=Austin,o=Tivoli,c=US sales
   cn=marketing,ou=Boston,o=Austin Sale,c=US marketing

Return Codes
The following exit status codes can be returned:
0  The command completed successfully.
The command failed. When a command fails, the **pdadmin** command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**

[group show](#)
**group modify**
Changes an existing group by adding a description, or adding or removing a list of members.

**Syntax**
```
padmin -a admin_id -p password group modify group_name (add (user...) | description description | remove (user...))
```

**Options**
- `-a admin_id` Specifies the administrator's ID. If this is the only option that is specified, the user is prompted for the password.
- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `group_name` Specifies the name of the group to be modified.
  
  A valid group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).
  
  Examples of group names: Credit, Sales, Test-group

- `add user...` Adds the specified users to the specified group. The format of the user list is a parenthesized list of user names, separated by spaces.
  
  For example, the specified users to be added might be: ("Mary Jones" dlucas mlucaser)

- `description description` Changes the description for the specified group.
  
  A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).
  
  If the description contains a space, ensure that you enclose the description in double quotation marks. You can specify an empty string (""") to clear an existing description.
  
  Example of description: "Credit, Dept HCUS"

- `remove user...` Removes the specified users from the specified group. The format of the user list is a parenthesized list of user names, separated by spaces.
  
  For example, the specified users to be deleted might be: ("Mary Jones" dlucas mlucaser)

**Examples**
1. The following example adds a new user `dlucas` to the `engineering` group:
   ```
padmin> group modify engineering add dlucas
   ```
2. The following example adds three new users to the `engineering` group:
   ```
padmin> group modify engineering add ("Mary Jones" dsmith mlucaser)
   ```
3. The following example deletes three existing users from the `engineering` group:
   ```
padmin> group modify engineering remove ("Mary Jones" dlucas mlucaser)
   ```
4. The following example changes the description of the credit group:

   pdadmin>group modify credit description "Credit, Dept HCUS"

**Return Codes**

The following exit status codes can be returned:

0  The command completed successfully.

1  The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**

- group create
- group import
group show
Shows the properties of the specified group.

Syntax
pdadmin –a admin_id –p password group {show group_name | show-dn dn | show-members group_name}

Options
–a admin_id Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

–p password Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.

show group_name Shows the properties of the group specified by group_name.

A valid group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of group names: Credit, Sales, Test-group

show-dn dn Shows the group specified by the group's identifier in the user registry. The returned group is defined in the user registry, but it is not necessarily a Tivoli Access Manager group. Groups that are not Tivoli Access Manager groups can be imported into Tivoli Access Manager by use of the pdadmin group import command.

The format for a distinguished name is similar to: "cn=Claude Wright,ou=Austin,o=Tivoli,c=us"

show-members group_name Lists the user names of the members of the specified group.

A valid group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of group names: Credit, Sales, Test-group

Examples
1. The following example displays properties of the credit group:
   pdadmin>group show credit

   Output is similar to the following:
   Group ID: credit
   LDAP dn: cn=credit,ou=Austin,o=Tivoli,c=US
   Description: Credit, Dept HCUS
   LDAP cn: credit
   Is SecGroup: true

2. The following example displays properties specified by the group's identifier in the user registry:
   pdadmin>group show-dn cn=credit,ou=Austin,o=Tivoli,c=US
Output is similar to the following:

Group ID: credit  
LDAP dn: cn=credit,ou=Austin,o=Tivoli,c=US  
Description: Credit, Dept HCUS  
LDAP cn: credit  
Is SecGroup: true

3. The following example lists the user names of the members of the **credit** group:

```
pdadmin> group show-members credit
```

Output is similar to the following:

```
dlucas
mlucaser
```

**Return Codes**

The following exit status codes can be returned:

0   The command completed successfully.
1   The command failed. When a command fails, the **pdadmin** command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**

[group list](#)
help
Obtains system help for pdadmin commands and options.

Syntax
pdadmin help \{topic | command\}

Options
topic Specifies the general command topic for which help is needed. Topics, listed in the same order they are displayed, are as follows:
• acl
• action
• object
• objectspace
• server
• rsr
• rsrccred
• rsrgroup
• admin
• login
• logout
• user
• group
• policy
• pop
• errtext

Miscellaneous commands are as follows:
• exit
• help
• quit

cmd Specifies the specific pdadmin command for which help is needed.

Examples
1. The following example lists commands specified by the action topic:
   help action

   Output is similar to the following:
   action create
   action delete
   action group list
   ...

2. The following example lists options available for the specified command:
   help action create

   Output is similar to the following:
   action create action_name action_label action_type
   action create action_name action_label action_type action_group_name
   ...
Return Codes

The following exit status codes can be returned:

0  The command completed successfully.
1  The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.
login

Establishes authentication credentials used when communicating with the Tivoli Access Manager policy server. These credentials are used to determine a user’s access privileges to policy server data. Most commands cannot be performed unless an explicit login is done. Unauthenticated users can only use the errtext, exit, help, login, logout and quit commands.

Syntax

`pdadmin login [-a admin_id] [-p password]`

Options

- `-a admin_id`  Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- `-p password`  Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

Description

Credentials are used to determine user access privileges to policy server data. With the exception of errtext, exit, help, login, logout, and quit commands, you cannot run a pdadmin command unless you login with a user ID and password.

Credentials are not accumulated or stacked. A login command completely replaces any existing credentials.

Examples

The following example logs the sec_master user into the policy server using the password `pa55w0rd`:

`pdadmin>login -a sec_master -p pa55w0rd`

Return Codes

The following exit status codes can be returned:

0  The command completed successfully.

1  The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`).

Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

exit / quit
logout
**logout**
Discards any authentication credentials that are in effect.

**Syntax**
padmin logout

**Options**
None.

**Examples**
The following example lists commands specified by the action topic:
```
padmin> logout
```

**Return Codes**
The following exit status codes can be returned:

0 The command completed successfully.

1 The command failed. When a command fails, the **padmin** command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**
exit / quit
login
object create

Creates a protected object.

Syntax

```
pdadmin -a admin_id -p password object create object_name description type ispolicyattachable {yes|no}
```

Options

- `-a admin_id` Specifies the administrator's ID. If this is the only option that is specified, the user is prompted for the password.

- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- `object_name` Specifies the name for the object being created. This name must be unique.

  A valid object name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of object names: `/Management/Groups/Travel`, `/WebSEAL`, and `/Management`

- `description` Specifies any text string describing the object being created.

  A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  If the description contains a space, ensure that you enclose the description in double quotation marks. You can specify an empty string ("") to clear an existing description.

  Example of description: "Travel Groups"

- `type` Specifies the type of object to be created. Types range from 0-17. For example, types 10 or 16 are appropriate for container objects.

  The type can be one of the following categories:
  - 0 – unknown
  - 1 – secure domain
  - 2 – file
  - 3 – executable program
  - 4 – directory
  - 5 – junction
  - 6 – WebSEAL server
  - 7 – unused
  - 8 – unused
  - 9 – HTTP server
  - 10 – nonexistent object
  - 11 – container object
  - 12 – leaf object
  - 13 – port

Chapter 1. pdadmin command line utility
14 – application container object
15 – application leaf object
16 – management object
17 – unused

ispolicyattachable {yes|no}  Specifies whether an ACL or a protected object policy can be attached to
this object. The only valid values are yes or no.

Examples
1. The following example creates the object named /Management/test-object that has a description of
Test Object and is an application container object. An ACL or a protected object policy can be
attached to this object:
   `pdadmin>object create /Management/test-object "Test Object" 14 ispolicyattachable yes`

2. The following example creates the object named /Management/Groups/Travel that has a description
of Travel Container Object and is an application container object. An ACL or a protected object policy
cannot be attached to this object:
   `pdadmin>object create /Management/Groups/Travel "Travel Container Object" 14 ispolicyattachable no`

Return Codes
The following exit status codes can be returned:

0  The command completed successfully.
1  The command failed. When a command fails, the pdadmin command provides a description of the
error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of
the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
`object delete`
object delete

Deletes a protected object.

Syntax

```
pdadmin -a admin_id -p password object delete object_name
```

Options

- `admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- `password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- `object_name` Specifies the protected object to be deleted.

A valid object name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of object names: /Management/Groups/Travel, /WebSEAL, and /Management

Examples

1. The following example deletes the object named /Management/test-object:
   ```
pdadmin> object delete /Management/test-object
   ```

2. The following example deletes the object named /Management/Groups/Travel:
   ```
pdadmin> object delete /Management/Groups/Travel
   ```

Return Codes

The following exit status codes can be returned:

- `0` The command completed successfully.
- `1` The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

- [object create](#)
object list

Lists any objects grouped under the specified protected object. Alternatively, lists all the extended attributes associated with the specified protected object.

Syntax

```
pdadmin –a admin_id –p password object list object_name [attribute]
```

Options

- `-a admin_id`  Specifies the administrator's ID. If this is the only option that is specified, the user is prompted for the password.
- `-p password`  Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `object_name`  Specifies the protected object.
  
  A valid object name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).
  
  Examples of object names: `/Management/Groups/Travel`, `/WebSEAL`, and `/Management`

- `attribute`  Lists all extended attributes associated with the protected object specified by the `object_name` option.

Examples

1. The following example lists the object named `/Management`

   ```
   object list /Management
   ```

   Displays a list similar to the following:
   
   `/Management/ACL`
   `/Management/Action`
   `/Management/Config`
   ...`
   `.Management/test-object`

2. The following example lists the object named `/Management/Groups/Travel` and also lists the extended attributes, which are set by modifying the object:

   ```
   object list /Management/test-object attribute
   ```

   Displays a list similar to the following:
   
   `test1`

Return Codes

The following exit status codes can be returned:

- `0`  The command completed successfully.
- `1`  The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`).

Refer to the `IBM Tivoli Access Manager Error Message Reference` This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.
See Also

- object listandshow
- object show
**object listandshow**

Lists any child objects grouped under the specified protected object and displays all values associated with each object.

**Syntax**

```
pdadmin -a admin_id -p password object listandshow object_name
```

**Options**

- `-a admin_id` Specifies the administrator's ID. If this is the only option that is specified, the user is prompted for the password.

- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- `object_name` Specifies the protected object for which the child objects and associated values are to be displayed.

A valid object name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of object names: /Management/Groups/Travel, /WebSEAL, and /Management

**Examples**

1. The following example lists the object named `/Management/test-object`

   ```
pdadmin>object listandshow /Management
   ```

   Displays information similar to the following:

   ```
   Name : /Management/test-object/
   Description : Test Object
   Type : <Application Container Object> : 14
   Is Policy Attachable : yes
   ```

2. The following example lists the object named `/Management/Groups/Travel` and also automatically lists extended attributes, if any:

   ```
pdadmin> object listandshow /Management/Groups
   ```

   Displays information similar to the following:

   ```
   Name : /Management/Groups/Travel
   Description : Travel Container Object
   Type : <Application Container Object> : 14
   Is Policy Attachable : no
   Extended Attributes :
     test1
     1111
   ```

**Return Codes**

The following exit status codes can be returned:

0 The command completed successfully.

1 The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x140c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#) This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.
See Also

object list
object show
**object modify**

Modifies an existing object.

**Syntax**

```
pdadmin -a admin_id -p password object modify object_name delete attribute attribute_name [attribute_value]
```

```
pdadmin -a admin_id -p password object modify object_name set attribute attribute_name attribute_value
```

```
pdadmin -a admin_id -p password object modify object_name set description description
```

```
pdadmin -a admin_id -p password object modify object_name set ispolicyattachable {yes|no}
```

```
pdadmin -a admin_id -p password object modify object_name set name new_object_name
```

```
pdadmin -a admin_id -p password object modify object_name set type type
```

**Options**

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- `object_name` Specifies the protected object to be modified.

  A valid object name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of object names: `/Management/Groups/Travel`, `/WebSEAL`, and `/Management`

- `delete attribute attribute_name [attribute_value]` Deletes the specified extended attribute (name and value) from the specified protected object. When you delete the last value for an attribute, it also deletes the attribute from the ACL.

  A valid extended attribute name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  The optional `attribute_value` deletes the specified value from the specified extended attribute key in the specified ACL. A valid extended attribute value is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of attribute names and values:

  | test1  | 1111 |
  | Dept_No | 445  |
  | Employee_name | "Diana Lucas" |
set attribute *attribute_name* *attribute_value*

Creates an extended attribute, with the specified name and value, and adds it to the specified protected object. If the attribute already exists, the attribute value is added as an additional value if the same value does not exist for this attribute. If the same value exists for this attribute, it does not get added again (duplicate values are not allowed), and no error is returned.

A valid attribute name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

The optional *attribute_value* deletes the specified value from the specified extended attribute key in the specified protected object. A valid attribute value is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of extended attribute names and values:

attr1 valueA
attr1 valueB
attr2 valueC

set description *description*

Sets the description field of the specified protected object.

A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

If the description contains a space, ensure that you enclose the description in double quotation marks. You can specify an empty string (""") to clear an existing description.

Example of description: "Travel Group aaa"

set ispolicyattachable {yes|no}

Sets whether the protected object can have an ACL or a POP attached or not. Valid values are yes or no.

set name *new_object_name*

Sets the name of the specified protected object.

set type *type*

Specifies the type of the object space to be created. Types range from 0-17. For example, types 10 or 16 are appropriate for objects.

The type can be one of the following categories:

  0 – unknown
  1 – secure domain
  2 – file
  3 – executable program
  4 – directory
  5 – junction
  6 – WebSEAL server
  7 – unused
  8 – unused
  9 – HTTP server
  10 – nonexistent object
  11 – container object
  12 – leaf object
  13 – port
14 – application container object
15 – application leaf object
16 – management object
17 – unused

Examples
1. The following example, entered on one line, sets the `ispolicyattachable` option:
   ```
   pdadmin>object modify /Management/Groups/Travel set ispolicyattachable yes
   ```
2. The following example sets attributes for the object `/Management/test-object`:
   ```
   pdadmin>object modify /Management/test-object set attribute test1 1111
   ```

Return Codes
The following exit status codes can be returned:

0  The command completed successfully.
1  The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
- `object create`
object show

Shows all values associated with the protected object. Alternatively,
returns the value associated with the specified extended attribute for the specified protected object. ACLs,
POPs, and authorization rules which are attached to the protected object.

Syntax

pdadmin –a admin_id –p password object show object_name [attribute attribute_name]

Options

–a admin_id Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

–p password Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.

object_name Returns the specified protected object.

A valid object name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of object names: /Management/Groups/Travel, /WebSEAL, and /Management.

attribute attribute_name Specifies the name of the extended attribute whose values are to be displayed.

A valid attribute name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of extended attribute names: Dept_No, Employee_Name.

Examples

1. The following example lists the object named /Management/test-objects, which also lists any extended attribute names and values:

   pdadmin>object show /Management/test-object

   Displays information similar to the following:

   Name : /Management/test-object/
   Description : Test Object
   Type : <Application Container Object> : 14
   Is Policy Attachable : yes

2. The following example, entered on one line, displays the object named /Management/test-objects and lists the extended attribute name and value for the attribute named test1:

   pdadmin>object show /Management/test-object attribute test1

   Displays information similar to the following:
Name : /Management/test-object/
  Description : Test Object
  Type : <Application Container Object> : 14
  Is Policy Attachable : yes
  Extended Attributes :
    test1
      1111

Return Codes
The following exit status codes can be returned:

0   The command completed successfully.
1   The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
object list
object listandshow
objectspace create

Creates a protected object space under which protected objects can be placed.

Syntax

```
pdadmin -a admin_id -p password objectspace create objectspace_name description type
```

Options

- **-a admin_id** Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- **-p password** Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.

- **objectspace_name** Specifies the name of the object space to be created.

  A valid object space name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of object space names: /Management, /WebSEAL

- **description** Specifies the description of the new object space.

  A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  If the description contains a space, ensure that you enclose the description in double quotation marks. You can specify an empty string (""") to clear an existing description.

  Example of description: "Accounting"

- **type** Specifies the type of the object space to be created. Types range from 0-17. For example, types 10 or 16 are appropriate for objects and object spaces.

  The type can be one of the following categories:
  0 – unknown
  1 – secure domain
  2 – file
  3 – executable program
  4 – directory
  5 – junction
  6 – WebSEAL server
  7 – unused
  8 – unused
  9 – HTTP server
  10 – nonexistent object
  11 – container object
  12 – leaf object
  13 – port
  14 – application container object
  15 – application leaf object
16 – management object
17 – unused

Description
The root of the new protected object space automatically has the `ispolicyattachable` attribute set to `true`.

Examples
1. The following example creates an object space named `/Test-Space` that is an application container object:
   ```
padmin> objectspace create /Test-Space "New Object Space" 14
   ```
2. The following example creates an object space named `/Dept4D4` that is a management object:
   ```
padmin> objectspace create /Dept4D4 "Department 4D4" 16
   ```

Return Codes
The following exit status codes can be returned:

0  The command completed successfully.
1  The command failed. When a command fails, the `padmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
`objectspace delete`
objectspace delete
Deletes the specified protected object space.

Syntax
```
pdadmin -a admin_id -p password objectspace delete objectspace_name
```

Options
- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `objectspace_name` Specifies the name of the object space to be deleted.

A valid object space name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of object space names: /Management, /WebSEAL

Examples
1. The following example deletes the object space named `/Test-Space`:
   ```
pdadmin> objectspace delete /Test-Space
   ```
2. The following example deletes the object space named `/Dept4D4`:
   ```
pdadmin> objectspace delete /Dept4d$Test-Space
   ```

Return Codes
The following exit status codes can be returned:

0 The command completed successfully.

1 The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
- `objectspace create`
objectspace list
Lists all of the existing protected object spaces in the policy server.

Syntax
pdadmin –a admin_id –p password objectspace list

Options
- `–a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `–p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

Examples
The following example lists all the protected object spaces:
pdadmin> objectspace list

Return Codes
The following exit status codes can be returned:
0 The command completed successfully.
1 The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.
policy get
Displays user password, account rules, and conditions.

Syntax

```
pdadmin -a admin_id -p password policy get account-expiry-date [-user user_name]
pdadmin -a admin_id -p password policy get disable-time-interval [-user user_name]
pdadmin -a admin_id -p password policy get max-login-failures [-user user_name]
pdadmin -a admin_id -p password policy get max-password-age [-user user_name]
pdadmin -a admin_id -p password policy get max-password-repeated-chars [-user user_name]
pdadmin -a admin_id -p password policy get min-password-alphas [-user user_name]
pdadmin -a admin_id -p password policy get min-password-length [-user user_name]
pdadmin -a admin_id -p password policy get min-password-non- alphas [-user user_name]
pdadmin -a admin_id -p password policy get password-spaces [-user user_name]
pdadmin -a admin_id -p password policy get tod-access [-user user_name]
```

Options

- **a admin_id**
  Specifies the administrator's ID. If this is the only option that is specified, the user is prompted for the password.

- **p password**
  Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.

account-expiry-date
Displays the account expiration date.

disable-time-interval
Displays the time, in seconds, to disable user accounts when the maximum number of login failures is exceeded.

max-login-failures
Displays the maximum number of login failures.

max-password-age
Displays the maximum time, in days expressed as 00–00:00:00, that a password will be valid. This time is relative to the last time the password was changed.

max-password-repeated-chars
Displays the maximum number of repeated characters allowed in a password.

min-password-alphas
Displays the minimum number of alphabetic characters required in a password.

min-password-length
Displays the minimum password length.

min-password-non-alphas
Displays the minimum number of non-alphabetic characters required in a password.
password-spaces
Displays whether spaces are allowed in passwords.

tod-access
Displays the time of day access policy.

–user user_name
Specifies the user whose policy information is to be displayed. If this option is not specified, the general policy is displayed. For any given policy, if a user has a specific policy applied, this specific policy takes precedence over any general policy that might also be defined. The precedence applies regardless of whether the specific policy is more or less restrictive than the general policy.

A valid user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of user names: dluca, sec_master, "Mary Jones"

Examples
1. The following example returns the account expiration date of unlimited for the specified user dluca:
   pdadmin>policy get account-expiry-date -user dluca
   Account expiry date: unlimited
2. The following example returns the maximum time of 0 days that the password is valid for the specified user dluca:
   pdadmin>policy get max-password-age -user dluca
   maximum password age: 0-0:0:0

Return Codes
The following exit status codes can be returned:

0  The command completed successfully.
1  The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the IBM Tivoli Access Manager Error Message Reference This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

policy set
policy set

Sets user password, account rules, and conditions.

Syntax

pdadmin –a admin_id –p password policy set account-expiry-date {unlimited|absolute_time|unset} [–user user_name]

pdadmin –a admin_id –p password policy set disable-time-interval {number|unset|disable} [–user user_name]

pdadmin –a admin_id –p password policy set max-login-failures number|unset [–user user_name]

pdadmin –a admin_id –p password policy set max-password-age {unset|relative_time} [–user user_name]

pdadmin –a admin_id –p password policy set max-password-repeated-chars number|unset [–user user_name]

pdadmin –a admin_id –p password policy set min-password-alphas {unset|number} [–user user_name]

pdadmin –a admin_id –p password policy set min-password-length {unset|number} [–user user_name]

pdadmin –a admin_id –p password policy set min-password-non-alphas {unset|number} [–user user_name]

pdadmin –a admin_id –p password policy set password-spaces {yes|no|unset} [–user user_name]

pdadmin –a admin_id –p password policy set tod-access {anyday|weekday|day_list}:{time_spec-time_spec} [:{utc|local}]:unset [–user user_name]

Options

– a admin_id
   Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for
   the password.

– p password
   Specifies the password for the admin_id user. If this option is not specified, the user is prompted
   for the password. The password cannot be specified if the admin_id is not specified.

account-expiry-date {unlimited|absolute_time|unset}
   Sets the account expiration date. The absolute_time format is specified as YYYY-MM-DD-
   hh:mm:ss. The hours must be entered using a 24-hour clock (for example, 09 for 9 a.m. or 14 for
   2 p.m.). The default value is unset.

   If you set the account expiration date, note that it will be set for all accounts that do not use the
   -user user_name option. By default, the sec_master user account has a per-user account
   expiration date of unlimited. If you set the account expiration date to unlimited, it is recommended
   that you also:
   • Set max-password-age to 0 for unlimited.
   • Set tod-access to anyday:anytime:local.
   • Use the -user user_name option.

disable-time-interval {number|unset|disable}
   Sets the time, in seconds, to disable each user account when the maximum number of login
   failures is exceeded. Tivoli Access Manager does not impose an upper limit for the maximum
number allowed. Use a range from 0 (unlimited) to a number that represents the value that is most logical for the parameter you are trying to set. The default value is 180.

**max-login-failures number/unset**

Sets the maximum number of login failures allowed. Tivoli Access Manager does not impose an upper limit for the maximum number allowed. Instead use a range from 0 to a number that represents the value that is most logical for the parameter you are trying to set. If the number is too large, it might render the login policy ineffective. The default value is 10.

**max-password-age {unset/relative_time}**

Sets the maximum time, in days, that a password will be valid. This policy is a global password policy as opposed to the individual `pdadmin user modify username password-valid` policy. The individual user policy enables or disables the validity of a password for the specified user account.

The `relative_time` option is relative to the number of days since the last password change occurred. The `relative_time` format is specified as `DDD-hh:mm:ss`. The valid range is from 000–00:00:00 to 999–23:59:59. The default value is 91 days, which is expressed as 91–00:00:00.

**max-password-repeated-chars number/unset**

Sets the maximum number of repeated characters allowed in a password. Tivoli Access Manager does not impose an upper limit for the maximum number allowed. Instead use a range from 0 to a number that represents the value that is most logical for the parameter you are trying to set. If the number is too large, it might render the password policy ineffective. The default value is 2.

**min-password-alphas {unset/number}**

Sets the minimum number of alphabetic characters required in a password. Tivoli Access Manager does not impose an upper limit for the minimum number allowed. Instead use a number that represents the value that is most logical for the parameter you are trying to set. If the number is too small, it might render the password policy ineffective. The default value is 4.

**min-password-length {unset/number}**

Sets the minimum password length. Tivoli Access Manager does not impose an upper limit for the minimum number allowed. Instead use a number that represents the value that is most logical for the parameter you are trying to set. If the number is too large, the password policy might be difficult to adhere to. The default value is 8.

**min-password-non-alphas {unset/number}**

Sets the minimum number of non-alphabetic characters required in a password. Tivoli Access Manager does not impose an upper limit for the minimum number allowed. Instead use a number that represents the value that is most logical for the parameter you are trying to set. If the number is too large, the password policy might be difficult to adhere to. The default value is 1.

**password-spaces {yes/no/unset}**

Sets the policy of whether spaces are allowed in passwords. The default value is unset.

**tod-access {\{anyday/weekday/day_list\};\{time_spec-time_spec\};\{utc/local\}\{unset\}}**

Sets the time of day access policy. The optional time zone is local by default. (Note: utc=GMT) The `time_spec` format is specified as `hh:mm` and is expressed using a 24-hour clock (for example, 0900 for 9 a.m. or 1430 for 2:30 p.m.). The default value is unset.

**–user user_name**

Specifies the user whose policy information is to be set. If this option is not specified, the general policy is set. For any given policy, if a user has a specific policy applied, this specific policy takes precedence over any general policy that might also be defined. The precedence applies regardless of whether the specific policy is more or less restrictive than the general policy.

A valid user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of user names: dlucas, sec_master, "Mary Jones"
Examples

1. The following example sets the account expiration date of December 30, 1999, at 11:30pm for the specified user dlucas:
   
   pdadmin>policy set account-expiry-date 1999-12-30-23:30:00 -user dlucas

2. The following example sets the maximum password age of 31 days 8 hours and 30 minutes for the specified user dlucas:
   
   pdadmin>policy set max-password-age 031-08:30:00 -user dlucas

Return Codes

The following exit status codes can be returned:

0  The command completed successfully.

1  The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

   Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

policy get
pop attach

Attaches a protected object policy (POP) to the specified protected object. The POP must be created before it can be attached.

Syntax

```
pdadmin -a admin_id -p password pop attach object_name pop_name
```

Options

- `-a admin_id` Specifies the administrator's ID. If this is the only option that is specified, the user is prompted for the password.
- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `object_name` Specifies the name of the protected object to which the protected object policy will be attached.
  
  A valid object name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of object names: `/Management/Groups/Travel`, `/WebSEAL`, and `/Management`

- `pop_name` Specifies the name of the protected object policy to be attached.

A valid POP name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of POP names: `poptest`, `pop1`

Description

At most, one POP can be attached to a given protected object. If the object already has a POP attached to it, the specified POP replaces the existing one. The same POP can be attached to multiple protected objects. Ensure that the protected object exists in the protect object space before attempting to attach a POP.

Examples

1. The following example attaches the POP `pop1` to the protected object named `/Management/test-object`:
   ```
pdadmin>pop attach /Management/test-object pop1
   ```
2. The following example attaches the POP `poptest` to the protected object named `/Test-Space`:
   ```
pdadmin>pop attach /Test-Space poptest
   ```

Return Codes

The following exit status codes can be returned:

0  The command completed successfully.
1  The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`).
Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

pop create
pop detach
**pop create**

Creates a protected object policy object.

**Syntax**

```
padmin -a admin_id -p password pop create pop_name
```

**Options**

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- `pop_name` Specifies the name of the protected object policy to be created.

A valid POP name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of POP names: `poptest`, `pop1`

**Examples**

The following example shows how to create and list a POP:

```
padmin>pop create test
```

The new POP contains new POP settings similar to the following:

```
padmin>pop show test
    Protected object policy: test
    Description: 
    Warning: no
    Audit Level: none
    Quality of protection: none
    Time of day access: sun, mon, tue, wed, thu, fri, sat: 
        anytime: local
    IP Endpoint Authentication Method Policy
        Any Other Network 0
```

**Return Codes**

The following exit status codes can be returned:

- `0` The command completed successfully.

- `1` The command failed. When a command fails, the `padmin` command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#) This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**

- `pop attach`
- `pop delete`
**pop delete**

Deletes the specified protected object policy.

**Syntax**

```
pdadmin -a admin_id -p password pop delete pop_name
```

**Options**

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- `pop_name` Specifies the name of the protected object policy to be deleted.

A valid POP name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of POP names: poptest, pop1

**Examples**

1. The following example deletes the POP `pop1`:
   ```
pdadmin>pop delete pop1
   ```
2. The following example deletes the POP `test`
   ```
pdadmin>pop delete poptest
   ```

**Return Codes**

The following exit status codes can be returned:

- **0** The command completed successfully.
- **1** The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](https://www.ibm.com/support/knowledgecenter/SG2GU7_6.4.0/com.ibm.oag.doc/pdadmin_error.html). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**

- `pop create`
- `pop detach`
pop detach
Detaches a protected object policy from the specified protected object.

Syntax
`pdadmin -a admin_id -p password pop detach object_name`

Options
- `a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `object_name` Specifies the protected object from which the protected object policy is to be detached.

A valid object name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of object names: /Management/Groups/Travel, /WebSEAL, and /Management

Examples
1. The following example detaches all POPs from the protected object named `/Management/test-object`:
   `pdadmin>pop detach /Management/test-object`
2. The following example detaches all POPs from the protected object named `/Test-Space`:
   `pdadmin>pop detach /Test-Space`

Return Codes
The following exit status codes can be returned:

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The command completed successfully.</td>
</tr>
<tr>
<td>1</td>
<td>The command failed. When a command fails, the <code>pdadmin</code> command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2). Refer to the <a href="#">IBM Tivoli Access Manager Error Message Reference</a>. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.</td>
</tr>
</tbody>
</table>

See Also
- pop attach
- pop delete
pop find
Finds and lists all protected objects that have protected object policies attached.

Syntax
```bash
pdadmin \-a \admin_id \-p password pop find pop_name
```

Options
- `-a \admin_id` Specifies the administrator's ID. If this is the only option that is specified, the user is prompted for the password.
- `-p password` Specifies the password for the \admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the \admin_id is not specified.
- `pop_name` Specifies the name of the protected object policy for which to search.

A valid POP name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of POP names: poptest, pop1

Examples
1. The following example finds all objects to which the POP `pop1` is attached:
   ```bash
   pdadmin> pop find pop1
   /Management/test-object
   ```
2. The following example finds all objects to which the POP `poptest` is attached:
   ```bash
   pdadmin> pop find poptest
   /Test-Space
   ```

Return Codes
The following exit status codes can be returned:

0  The command completed successfully.
1  The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
- `pop list`
**pop list**

Lists all protected object policies that have been created. Alternatively, lists all extended attributes associated with a protected object policy.

**Syntax**

```
padmin -a admin_id -p password pop list [pop_name] [attribute]
```

**Options**

- **-a admin_id** Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- **-p password** Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.
- **pop_name** Specifies the POP for which to list the attributes.

A valid POP name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of POP names: poptest, pop1

**Examples**

1. The following example shows how to list all POPs:
   ```
   pdadmin> pop list
   test
   pop1
   poptest
   ```

2. The following example shows how to list all the attributes for the POP **pop1**:
   ```
   pdadmin> pop list pop1 attribute
   attr1
   ```

**Return Codes**

The following exit status codes can be returned:

- **0** The command completed successfully.
- **1** The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**

[pop find](#)
pop modify

Modifies protected object policies.

Syntax

```
pdadmin -a admin_id -p password pop modify pop_name delete attribute attribute_name [attribute_value]
pdadmin -a admin_id -p password pop modify pop_name set attribute attribute_name attribute_value
pdadmin -a admin_id -p password pop modify pop_name set audit-level {all|none|permit|deny|audit_level_list}
pdadmin -a admin_id -p password pop modify pop_name set description description
pdadmin -a admin_id -p password pop modify pop_name set ipauth add network netmask authority_level
pdadmin -a admin_id -p password pop modify pop_name set ipauth anyothernw authority_level
pdadmin -a admin_id -p password pop modify pop_name set ipauth remove network netmask
pdadmin -a admin_id -p password pop modify pop_name set pop {none|integrity|privacy}
pdadmin -a admin_id -p password pop modify pop_name set tod-access {anyday|weekday|day_list}:{anytime|time_spec-time_spec}[:{utc|local}]
pdadmin -a admin_id -p password pop modify pop_name set warning {yes|no}
```

Options

- **-a admin_id**
  Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- **-p password**
  Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.

**delete attribute attribute_name [attribute_value]**

Deletes the specified value from the specified extended attribute key in the specified POP.

A valid attribute name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

The optional attribute_value deletes the specified value from the specified extended attribute key in the specified POP.

Examples of extended attribute names and values:

<table>
<thead>
<tr>
<th>Dept_No</th>
<th>445</th>
</tr>
</thead>
<tbody>
<tr>
<td>Employee_Name</td>
<td>&quot;Diana Lucas&quot;</td>
</tr>
</tbody>
</table>

**pop_name**

 Specifies the name of the protected object policy to be modified.

**set attribute attribute_value**

Sets or modifies the specified value from the specified extended attribute key in the specified POP.
If the attribute already exists, the attribute value is added as an additional value if the same value does not exist for this attribute. If the same value exists for this attribute, it does not get added again (duplicate values are not allowed), and no error is returned.

A valid attribute name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

The optional attribute_value deletes the specified value from the specified extended attribute key in the specified POP.

Example: "attribute_value" deletes the specified value from the specified extended attribute key in the specified POP.

**Examples**

1. This example shows how to modify a POP description:
   ```bash
   pdadmin>pop modify test description "Test POP"
   ```

2. This example shows how to turn the warning mode on:
   ```bash
   pdadmin>pop modify test set warning yes
   ```
3. This example shows how to set the audit level to audit all requests on a protected object that result in successful access (permit) and in denial of access (deny).

   pdadmin>pop modify test set audit-level permit,deny

4. This example shows how to set an attribute named attr1 with a value of valueA for the POP named pop1:

   pdadmin>pop modify pop1 set attribute attr1 valueA

Return Codes

The following exit status codes can be returned:

0  The command completed successfully.

1  The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

   Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

   pop attach
   pop create
**pop show**

Shows details of the protected object policy. Alternatively, displays the values for the specified extended attribute from the specified protected object policy.

**Syntax**

```
pdadmin -a admin_id -p password pop show pop_name [attribute ]
```

**Options**

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `pop_name` Specifies the POP to display.
  A valid POP name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).
  Examples of POP names: poptest, pop1
- `attribute attribute_name` Specifies the name of the extended attribute whose values you want to display.
  A valid attribute name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).
  Examples: Dept_No, Employee_Name

**Examples**

1. The following example shows how to show POP information, including the description:

   ```
pdadmin>pop show test
   Protected object policy: test
   Description: Test POP
   Warning: no
   Audit level: none
   Quality of protection: none
   Time of day access: sun, mon, tue, wed, thu, fri, sat:
     anytime: local
   IP Endpoint Authentication Method Policy
   Any Other Network 0
   ```

2. The following example shows attribute `attr1` information for the POP `pop1`:

   ```
pdadmin>pop show pop1 attribute attr
   attr1
   valueA
   ```

**Return Codes**

The following exit status codes can be returned:

- `0` The command completed successfully.
The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**

- pop find
- pop list
rsrcreg

Creates and names a server as a resource.

Syntax

```
pdadmin -a admin_id -p password rsrcreg resource_name [-desc description]
```

Options

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- `resource_name` Specifies the name of the resource to be created.

  A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of resource names: engwebs01JonesData

- `-desc description` Specifies a description for the resource. Descriptions containing a space must be enclosed in double quotes.

  A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  If the description contains a space, ensure that you enclose the description in double quotation marks. You can specify an empty string ("") to clear an existing description.

  Examples of descriptions: "Engineering Web server – Room 4807", "Printer in room 345, Bldg 2"

Description

A Web resource is a Web server that serves as the backend of a WebSEAL junction.

Examples

1. The following example, entered as one line, creates and names a Web resource engwebs01 with an associated description "Engineering Web server – Room 4807":

   ```
pdadmin>rsrc create engwebs01 –desc "Engineering Web server – Room 4807"
   
   "Engineer in in room 4807"
   ```

2. The following example, entered as one line, creates and names a printer resource with an associated description "Printer in room 345, Bldg 2":

   ```
pdadmin>rsrc create "Mary Jones Printer" –desc "Printer in room 345, Bldg 2"
   
   "Printer in room 345, Bldg 2"
   ```

Return Codes

The following exit status codes can be returned:
0  The command completed successfully.
1  The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#) This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

`rsrc delete`
rsrc delete

Deletes the specified single signon resource.

Syntax

```
pdadmin –a admin_id –p password rsrc delete resource_name
```

Options

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `resource_name` Specifies the name of the resource to be deleted. The resource must exist, or an error is displayed.

A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of resource names: engwebs01JonesData

Examples

1. The following example deletes the named resource `engwebs01`:
   
   `pdadmin>rsrc delete engwebs01`

2. The following example deletes the named resource "Mary Jones Printer" :
   
   `pdadmin>rsrc delete "Mary Jones Printer"`

Return Codes

The following exit status codes can be returned:

0  The command completed successfully.
1  The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

rsrc create
rsrc list

Returns a list of all the single signon resource names.

Syntax

```
pdadmin –a admin_id –p password rsrc list
```

Options

- `–a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- `–p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

Examples

The following example returns a list of all the single signon Web resource names:

```
pdadmin>rsrc list
```

Output is similar to the following:

```
engwebs01
Mary Jones Printer
```

Return Codes

The following exit status codes can be returned:

- **0** The command completed successfully.

- **1** The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

`rsrc create`
rsrsrc show
Displays the resource information for the named resource.

Syntax
```
   pdadmin -a admin_id -p password rsrc show resource_name
```

Options
- `a admin_id` Specifies the administrator's ID. If this is the only option that is specified, the user is prompted for the password.
- `p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `resource_name` Specifies the name of the resource for which information is shown. The resource must exist, or an error is displayed.

A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of resource names: engwebs01JonesData

Examples
1. The following example returns information for the specified resource `engwebs01`:
   ```
   pdadmin>rsrc show engwebs01
   ```
   Output is similar to the following:
   ```
   Web Resource Name: engwebs01
   Description: Engineering Web server - Room 4807
   ```
2. The following example returns information for the specified resource "Mary Jones Printer":
   ```
   pdadmin>rsrc show "Mary Jones Printer"
   ```
   Output is similar to the following:
   ```
   Web Resource Name: Mary Jones Printer
   Description: Printer in room 345, Bldg 2
   ```

Return Codes
The following exit status codes can be returned:

0 The command completed successfully.
1 The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
```
rsrsrc list
```

82 IBM Tivoli Access Manager: Command Reference
rsrccred create

Creates a single signon credential.

Syntax

```
padmin -a admin_id -p password rsrccred create resource_name rsrcreuser resource_userid rsrcrepwd resource_password rsrcttype {web|group} user user_name
```

Options

- **–a admin_id**
  Specifies the administrator's ID. If this is the only option that is specified, the user is prompted for the password.

- **–p password**
  Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- **resource_name**
  Specifies the name given to the resource when the resource was created. The resource (or resource group) must already exist in order to create the resource credential. If the resource (or resource group) does not exist or is not specified, an error message is displayed.

  A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of resource names: engwebs01"Mary Jones Printer"

- **rsrcreuser resource_userid**
  Specifies the unique user identification (user ID) for the user at the Web server.

  A valid user identification is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of user identifications: 4807ws01, userD4D

- **rsrcrepwd resource_password**
  Specifies the password for a user at the Web server.

- **rsrcttype {web|group}**
  Specifies whether the resource type is web or group.

- **user user_name**
  Specifies the name of the user for whom the resource credential information applies. If the user does not exist or is not specified, an error message is displayed.

  A valid user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of user names: dlucas, sec_master, "Mary Jones"

Examples

1. The following example, entered on one line, creates the Web resource credential named engwebs01 for the resource user ID 4807ws01 and password pwd4lucas given to user dlucas:
pdadmin>rsrccred create engwebs01 rsrccuser \  
4807ws01 rsrccpwd pwd4lucas rsrctype web user dluca

2. The following example, entered on one line, creates the group resource credential named 
   printerusers for the resource user ID userD4D and password pwd4mjones given to user "Mary Jones":
   pdadmin>rsrccred create printerusers rsrccuser \  
   userD4D rsrccpwd pwd4mjones rsrctype group user "Mary Jones"

**Return Codes**

The following exit status codes can be returned:

0  The command completed successfully.

1  The command failed. When a command fails, the `pdadmin` command provides a description of the 
   error and an error status code in hexadecimal format (for example, 0x14c012f2).

   Refer to the [IBM Tivoli Access Manager Error Message Reference](#) This reference provides a list of 
   the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**

rsrcc delete
rsrccred delete

Deletes a single signon credential.

Syntax

    pdadmin -a admin_id -p password rsrccred delete resource_name rsrctype {web|group} user user_name

Options

- `a admin_id` Specifies the administrator's ID. If this is the only option that is specified, the user is prompted for the password.

- `p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- `resource_name` Specifies the name given to the resource when the resource was created.

  A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of resource names: engwebs01"Mary Jones Printer"

- `rsrctype {web|group}` Specifies whether the resource type is web or group. The type of resource must match the resource type assigned when the resource was first created.

- `user user_name` Specifies the name of the user for whom the resource credential information applies.

  A valid user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of user names: dlucas, sec_master, "Mary Jones"

Examples

1. The following example deletes the resource credential information for the given resource engwebs01, resource type web, and user name dlucas:

    pdadmin>rsrccred delete engwebs01 rsrctype web user dlucas

2. The following example deletes the resource credential information for the given resource printerusers, resource type group, and user name "Mary Jones":

    pdadmin>rsrccred delete printerusers rsrctype group user "Mary Jones"

Return Codes

The following exit status codes can be returned:

- **0** The command completed successfully.

- **1** The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).
Refer to the [IBM Tivoli Access Manager Error Message Reference]. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
rsccred create
rsrccred list user

Returns the list of single signon credentials for the specified user.

Syntax

```
padmin -a admin_id -p password rsrccred list user user_name
```

Options

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `user_name` Specifies the name of the user for whom the resource credential information applies.

A valid user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of user names: dluca, sec_master, "Mary Jones"

Examples

The following example returns the list of single signon credentials for the specified user:

```
padmin>rsrccred list user dluca
```

Output is similar to the following:

Resource name: engwebs01
Resource Type: web

Return Codes

The following exit status codes can be returned:

- `0` The command completed successfully.
- `1` The command failed. When a command fails, the `padmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the `IBM Tivoli Access Manager Error Message Reference`. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.
rsrccred modify
Changes a single signon credential.

Syntax

```plaintext
pdadmin –a admin_id –p password rsrccred modify resource_name rsrctype {web|group}
set [–rsrcuser new_resource_userid] [–rsrcpwd new_resource_password] user user_name
```

Options

- **–a admin_id**
  Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- **–p password**
  Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- **resource_name**
  Specifies the name given to the resource when the resource was created.
  A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).
  Examples of resource names: engwebs01"Mary Jones Printer"

- **rsrctype {web|group}**
  Specifies whether the resource type is `web` or `group`. The type of resource must match the resource type assigned when the resource was first created.

- **–rsrcuser new_resource_userid**
  Specifies the new unique user identification (user ID) for the user at the Web server. To change or reset the resource user ID of the user, this optional command must be preceded by a dash (–).
  A valid user identification is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).
  Examples of user identifications: , 4807ws01, userD4D

- **–rsrcpwd new_resource_password**
  Specifies the new password for a user at the Web server. To change or reset the password information, this optional command must be preceded by a dash (–). Specifying this option without specifying the `–rsrcuser` option clears both the resource user ID and the resource password from the resource credential. To set the resource password, you must specify both the resource user ID and the resource password.

- **user user_name**
  Specifies the name of the user for whom the resource credential information applies.
  A valid user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).
  Examples of user names: dlucas, sec_master, "Mary Jones"
Examples

1. The following example, entered as one line, modifies the password to newrsrpw for the specified resource engwebs01:
   
   ```
   pdadmin>rsrccred modify engwebs01 rsrctype web \
   set -rsrcuser 4807ws01 -rsrcpwd newrsrpw user dlucas
   ```

2. The following example, entered as one line, modifies the resource user ID to user888 for the specified resource printerusers:
   
   ```
   pdadmin>rsrccred modify printerusers rsrctype group \
   set -rsrcuser user888 user "Mary Jones"
   ```

Return Codes

The following exit status codes can be returned:

0  The command completed successfully.

1  The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

rsrccred create
rsrccred show
Displays the attributes of a single signon credential. The credential identifier is composed of a resource name, a resource type, and a user name.

Syntax
```
padmin -a admin_id -p password rsrccred show resource_name rsrctype {weblgroup} user user_name
```

Options
- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `-p password` Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.
- `resource_name` Specifies the name of the single signon resource associated with the credential.
  
  A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of resource names: engwebs01, printerusers

- `rsrctype {weblgroup}` Specifies whether the resource type is web or group for the single signon resource associated with the credential.

- `user user_name` Specifies the name of the user associated with this credential.

  A valid user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of user names: dluca, sec_master, "Mary Jones"

Examples
1. The following example displays the specified single signon credential:
   ```
padmin>rsrccred show engwebs01 rsrctype web user dluca
   ```
   Output is similar to the following:
   
   Resource Name: engwebs01
   Resource Type: web
   Resource User Id: dluca

2. The following example displays the specified single signon credential:
   ```
padmin>rsrccred show user888 rsrctype group user "Mary Jones"
   ```
   Output is similar to the following:
   
   Resource Name: printerusers
   Resource Type: group
   Resource User Id: Mary Jones
**Return Codes**

The following exit status codes can be returned:

0   The command completed successfully.

1   The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

   Refer to the *IBM Tivoli Access Manager Error Message Reference*. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**

rsrccred list user
rsrgroup create
Creates and names a resource group.

Syntax
pdadmin –a admin_id –p password rsrgroup create resource_group_name [–desc description]

Options
–a admin_id Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
–p password Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.
resource_group_name Specifies the name of the resource group.
A valid resource group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).
Example of resource group names: webs4807, engwebs01, IBMprinters
–desc description Optional, specifies a description to identify this resource group.
A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).
If the description contains a space, ensure that you enclose the description in double quotation marks. You can specify an empty string (""") to clear an existing description.
Examples of descriptions: "Engineering Web server – Room 4807", "Printer in room 345, Bldg 2"

Examples
The following example creates and names a Web resource group and provides a description for that resource:

```
pdadmin>rsrgroup create webs4807 –desc "Web servers, Room 4807"
```

Return Codes
The following exit status codes can be returned:

0 The command completed successfully.
1 The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
rsrgroup delete
**rsrcgroup delete**

Deletes a single signon resource group.

**Syntax**

```
pdadmin -a admin_id -p password rsrcgroup delete resource_group_name
```

**Options**

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

`resource_group_name` Specifies the name of the resource group. The resource group must exist.

A valid resource group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

**Examples**

The following example deletes the named resource group and its associated description information:

```
padmin>rsrcgroup delete webs4807
```

**Return Codes**

The following exit status codes can be returned:

- **0** The command completed successfully.
- **1** The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**

[rsrcgroup create](#)
rsrgroup list
Displays the names of all resource groups defined in the user registry.

Syntax
pdadmin –a admin_id –p password rsrgroup list

Options
– a admin_id Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

– p password Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.

Examples
The following example returns a list of all single signon resource group names:

```
pdadmin>rsrcgroup list
```

Output is similar to the following:

```
webs4807
websbld3
```

Return Codes
The following exit status codes can be returned:

0 The command completed successfully.
1 The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#) This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
rsrgroup show
rsrcgroup modify

Adds or removes a single signon resource to or from a single signon resource group.

Syntax

```
pdadmin -a admin_id -p password rsrcgroup modify resource_group_name add rsrcname resource_name
```

```
pdadmin -a admin_id -p password rsrcgroup modify resource_group_name remove rsrcname resource_name
```

Options

- `-a admin_id`
  Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- `-p password`
  Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

`resource_group_name`

Specifies the name of the resource group to be modified.

A valid resource group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Example of resource group names: webs4807, engwebs01, IBMprinters

- `add rsrcname resource_name`
  Adds a single signon resource to the specified single signon resource group.

  A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of resource names: engwebs01"Mary Jones Printer"

- `remove rsrcname resource_name`
  Removes a single signon resource from the specified single signon resource group.

  A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of resource names: engwebs01"Mary Jones Printer"

Examples

1. The following example adds the named resource to the existing Web resource group:

   ```
   pdadmin>rsrgroup modify webs4807 add rsrcname engwebs02
   ```

2. The following example deletes the named resource from the existing Web resource group:

   ```
   pdadmin>rsrgroup modify webs4807 remove rsrcname engwebs02
   ```

Return Codes

The following exit status codes can be returned:
0 The command completed successfully.

1 The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#) This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**

`rsrcgroup create`
rsrgroup show
Displays the resource group information for the specified resource group.

Syntax
pdadmin –a admin_id –p password rsrgroup show resource_group_name

Options
–a admin_id  Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
–p password  Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.
resource_group_name  Specifies the name of the resource group.

Description
A valid resource group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).
Example of resource group names: webs4807, engwebs01, IBMprinters

Examples
The following example returns the specified single signon resource group:
pdadmin>rsrcgroup show webs4807

Output is similar to the following:
Resource Group Name: webs4807
Description: Web servers, Room 4807
Resource Members:
engwebs01
engwebs02
engwebs03

Return Codes
The following exit status codes can be returned:
0  The command completed successfully.
1  The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).
Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
rsrgroup list
server list
Lists all registered servers.

Syntax
pdadmin -a admin_id -p password server list

Options
- a admin_id Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- p password Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.

Description
pdadmin server commands require a server_name option. This option must be entered in the exact format as displayed in the output of the pdadmin server list command.

The server_name option is the full expression of the actual machine name and the Tivoli Access Manager component used by this command (such as WebSEAL). The Tivoli Access Manager server_name option is in the format:
access_manager_component-machine_name

For example, if the machine name is cruz and the Tivoli Access Manager component is WebSEAL, the server_name is:
webseald-cruz

Examples
The following example lists all registered servers:
pdadmin>server list

Output is similar to the following:
ivacld-topserver
ivacld-server2
ivacld-server3
ivacld-server4

Return Codes
The following exit status codes can be returned:
0 The command completed successfully.
1 The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
server list
tasks
server show
server listtasks

Retrieves the list of tasks (commands) available for this server.

Syntax

```
padmin –a admin_id –p password server listtasks server_name
```

Options

`–a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

`–p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

`server_name` Specifies the name of the Tivoli Access Manager server for which available tasks (commands) will be listed.

A valid Tivoli Access Manager server name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

The `server_name` option is the full expression of the actual machine name and the Tivoli Access Manager component used by this command (such as WebSEAL). The `server_name` option is in the format:

`access_manager_component-machine_name`

For example, if the machine name is `cruz` and the Tivoli Access Manager component is WebSEAL, the `server_name` is: `webseald-cruz`

Examples

The following example displays the list of available tasks from the server:

```
padmin>server listtasks ivacld-mogman.admogman.com
```

Output is similar to the following:

```
trace set component level [file path=file|other-log-agent-config]
trace show [component]
trace list [component]
stats show [component]
stats list
stats on [component] [interval] [count] [file path= file|other-log-agent-config]
stats off [component]
stats reset [component]
stats get [component]
```

Return Codes

The following exit status codes can be returned:

0  The command completed successfully.

1  The command failed. When a command fails, the `padmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.
See Also

server list
server show
server replicate

Notifies the authorization servers to receive database updates.

Syntax

```
pdadmin -a admin_id -p password server replicate [-server server_name]
```

Options

- `–a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `–p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `–server server_name` Specifies the name of the Tivoli Access Manager server to receive database updates. If `server_name` is specified but is not configured to receive database updates, an error message is displayed. If `server_name` is not specified, all servers configured to receive updates are notified.

A valid Tivoli Access Manager server name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

The `server_name` option is the full expression of the actual machine name and the Tivoli Access Manager component used by this command (such as WebSEAL). The `server_name` option is in the format:

```
access_manager_component-machine_name
```

For example, if the machine name is `cruz` and the Tivoli Access Manager component is WebSEAL, the `server_name` is: `webseald-cruz`

Examples

The following is an example of this command when specifying the `server_name`:

```
pdadmin>server replicate -server ivacld-topserver
```

Return Codes

The following exit status codes can be returned:

- **0** The command completed successfully.
- **1** The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`).

Refer to the `IBM Tivoli Access Manager Error Message Reference`. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.
server show

Displays the specified server's properties.

Syntax

```plaintext
pdadmin –a admin_id –p password server show server_name
```

Options

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `server_name` Specifies the name of the Tivoli Access Manager server whose properties are to be displayed.

A valid Tivoli Access Manager server name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

The `server_name` option is the full expression of the actual machine name and the Tivoli Access Manager component used by this command (such as WebSEAL). The `server_name` option is in the format:

```plaintext
access_manager_component-machine_name
```

For example, if the machine name is `cruz` and the Tivoli Access Manager component is WebSEAL, the `server_name` is: `webseald-cruz`

Examples

1. The following example displays the specified server's properties:

   ```plaintext
   pdadmin>server show ivacld-myhost
   ```

   Output is similar to the following:

   ```plaintext
   ivacld-myhost
   Description: ivacld/myhost
   Hostname: myhost
   Principal: ivacld/myhost
   Administration Request Port: 7137
   Listening for authorization database update notifications: yes
   AZN Administration Services:
     AZN_ADMIN_SVC_TRACE
   ```

2. The following example displays the properties of the WebSEAL server on the `cruz` machine:

   ```plaintext
   pdadmin>server show webseald-cruz
   ```

   Output is similar to the following:

   ```plaintext
   webseald-cruz
   Description: webseald/cruz
   Hostname: cruz
   Principal: webseald/cruz
   Port: 7234
   Listening for authorization database update notifications: yes
   AZN Administration Services:
     webseal-admin-svc
     azn_admin_svc_trace
   ```
Return Codes

The following exit status codes can be returned:

0  The command completed successfully.
1  The command failed. When a command fails, the **pdadmin** command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the *IBM Tivoli Access Manager Error Message Reference*. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

[server list](#)
server task

Purpose
Sends a command to a registered server.

Syntax
pdadmin -a admin_id -p password server task server_nameserver_task

Options
-a admin_id Specifies the administrator's ID. If this is the only option that is specified, the user is prompted for the password.

-p password Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.

server_name Specifies the name of the Tivoli Access Manager server. You must specify the server name in the exact format as displayed in the output of the pdadmin server list command.

A valid Tivoli Access Manager server name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

The server_name option is the full expression of the actual machine name and the Tivoli Access Manager component used by this command (such as WebSEAL). The server_name option is in the format:

access_manager_component-machine_name

For example, if the machine name is cruz and the Tivoli Access Manager component is WebSEAL, the server_name is: webseal-cruz

server_task Specifies the task (command) being sent.

Return Codes
The following exit status codes can be returned:

0 The command completed successfully.

1 The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
server task (WebSEAL)
server task add (WebSEAL)
server task stats (WebSEAL)
server task trace
server task (WebSEAL)

Creates and manipulates WebSEAL junctions. This command is available only if you have the Tivoli Access Manager WebSEAL product installed.

Syntax

```plaintext
pdadmin –a admin_id –p password server task server_name-host_name [ add options | create options | delete | exit | help [command]] jmt {load | clear} | list | remove | show] junction_point
```

Options

- `–a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `–p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `server_name-host_name` Specifies the name of the Tivoli Access Manager server.

For a single WebSEAL server, `server_name` is `webseal`. For multiple WebSEAL instances on the same machine, `server_name` is the configured name of the WebSEAL server instance followed by `–webseal`. For example, if the configured name of a WebSEAL instance is `webseal2`, the `server_name` is as follows: `webseal2–webseal`.

**Note:** The initial WebSEAL server installed on a machine is always named after the machine. For example, if the machine name is `patton`, the `host_name` is `patton`.

- `add options` Adds a server to an existing WebSEAL junction point. For more information, see “server task add (WebSEAL)” on page 107.
- `create options` Creates a new junction for an initial server. For more information, see the [IBM Tivoli Access Manager WebSEAL Administrator’s Guide](#).
- `delete` Removes the junction point specified by `junction_point`.
- `exit` Exits the `pdadmin` command line interface.
- `help [command]` Lists junction commands. If the `command` option is specified, lists detailed help for a specific junction command.
- `jmt {load | clear}` Loads or clears junction mapping table data, located in the `jmt.conf` file.
- `list` Lists all junction points on this server.
- `remove` Removes the specified server from a junction point.
- `show` Displays details of a junction.
- `junction_point` Specifies the junction point.

Authorization

`sec_master` administrative user

Return Codes

The following exit status codes can be returned:

0 The command completed successfully.
The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**

- `server task`  
- `server task stats (WebSEAL)`  
- `server task trace`  
- `server task add (WebSEAL)`
server task add (WebSEAL)

Adds an additional server to an existing junction point. This command is available only if you have the Tivoli Access Manager WebSEAL product installed.

Syntax
For TCP and SSL proxy junctions:

```
pdadmin –a admin_id –p password server task server_name-host_name add –h host_name –H host_name –P port [–D dn –i –p port –q url –u uuid –v virt_host_name –w] junction_point
```

For local, TCP, and SSL junctions:

```
pdadmin –a admin_id –p password server task server_name-host_name add –h host_name [–D dn –i –p port –q url –u uuid –v virt_host_name –w] junction_point
```

Options

- `–a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `–p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `server_name-host_name` Specifies the name of the Tivoli Access Manager server.
  - For a single WebSEAL server, `server_name` is `webseald`. For multiple WebSEAL instances on the same machine, `server_name` is the configured name of the WebSEAL server instance followed by `–webseald`. For example, if the configured name of a WebSEAL instance is `webseal2`, the `server_name` is as follows: `webseal2–webseald`.
  - Note: The initial WebSEAL server installed on a machine is always named after the machine. For example, if the machine name is `patton`, the `host_name` is `patton`.
- `–D dn` Specifies the distinguished name of the back-end server certificate. This value, matched with actual certificate DN enhances authentication.
  - A distinguished name for a server might be similar to: `cn=ivacld/libra,cn=SecurityDaemons,secAuthority=Default`
- `–H host_name` Specifies the DNS host name or IP address of the proxy server.
  - Valid values for `host_name` include any valid IP host name.

Examples:

```
host = libra
host = libra.dallas.ibm.com
```

- `–i` WebSEAL server treats as case insensitive.
- `–P port` Specifies the TCP port of the proxy server.
  - For `port_number`, use any valid port number. A valid port number is any positive number that is allowed by TCP/IP and that is not currently being used by another application. It is recommended that you use the default port number value, or else use a port number over 1000 currently not being used. For example: 6139
-p port Specifies the TCP port of the back-end third party server. The default value is 80 for TCP junctions; 443 for SSL junctions.

-q url Specifies the relative path for the query_contents script. By default, Tivoli Access Manager looks for query_contents in /cgi_bin/. If this directory is different or the query_contents file name is renamed, use this option to indicate to WebSEAL the new URL to the file.

-u uuid Specifies the UUID of a back-end server connected to WebSEAL using a stateful junction (–s).

-v virt_host_name Specifies the virtual host name represented on the back-end server. This option supports a virtual host setup on the back-end server. Use –v when the back-end junction server expects a host name header because you are junctioning to one virtual instance of that server. The default HTTP header request from the browser does not know that the back-end server has multiple names and multiple virtual servers. You must configure WebSEAL to supply that extra header information in requests destined for a back-end server set up as a virtual host.

-w Specifies Win32 file system support.

Authorization
sec_master administration user

Return Codes
The following exit status codes can be returned:

0 The command completed successfully.

1 The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
server show
server task stats (WebSEAL)
server task trace
server task (WebSEAL)
server task stats (WebSEAL)

Enables the gathering of statistical information. This command is available only if you have the Tivoli Access Manager WebSEAL product installed.

Syntax

```
pdadmin –a admin_id –p password server task server_name–host_name stats {get | list | off | reset | show} [component]
```

```
pdadmin –a admin_id –p password server task server_name–host_name stats on component [interval [count]] [log_agent]
```

Options

`–a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

`–p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

```
server_name–host_name
```

Specifies the name of the Tivoli Access Manager server.

For a single WebSEAL server, `server_name` is `webseald`. For multiple WebSEAL instances on the same machine, `server_name` is the configured name of the WebSEAL server instance followed by `-webseald`. For example, if the configured name of a WebSEAL instance is `webseal2`, the `server_name` is as follows: `webseal2–webseald`.

**Note:** The initial WebSEAL server installed on a machine is always named after the machine. For example, if the machine name is `patton`, the `host_name` is `patton`.

```
stats get [component]
```

Displays the current values of statistics being gathered for all enabled components. Specify the `component` option to get statistics for a specific enabled component.

```
stats list [component]
```

Lists all components available to gather and report statistics. Specify the `component` option to list a specific enabled component. If the specified component is not enabled, no output is displayed.

```
stats on component [interval [count]] [log_agent]
```

Enables or disables statistics dynamically statistics gathering for the specified component. When enabling stats, you can also set the statistics report frequency, count, and destination for a component. Options are as follows:

```
interval
```

Specifies the time interval between reports of information. This results in statistics being sent to a log file. When this option is specified, statistics are sent, by default, to standard out of the WebSEAL server, which is the WebSEAL log file. You can specify another output location using the `logagent` argument. If interval is not specified, no statistics are sent to any log file. However, the statistic component is still enabled. You can obtain reports dynamically at any time using the `pdadmin stats get` command.

```
count
```

Specifies the number of reports sent to a log file. The `interval` option is required if using the `count` option. If `interval` is specified without `count`, the duration of reporting is indefinite. After the count value is reached, reporting to a log file stops. However, the statistic component is still enabled. You can obtain reports dynamically at any time using the `pdadmin stats get` command.
log_agent Specifies a destination for the statistics information gathered for the specified component. For more information about event logging, see the IBM Tivoli Access Manager Base Administrator's Guide.

stats off [component] Disables statistics gathering for all components. Specify the component option to disable statistics gathering for a specific enabled component.

Note: By default, the pdweb threads, pdweb doccache, and pdweb jmt components are always enabled and cannot be disabled.

stats reset [component] Resets the values being gathered by all enabled components. Specify the component option to reset the values for a specific enabled component.

stats show [component] Shows the names and levels for all enabled stats components. Specify the component option to show the name and level for a specific enabled component.

Description For more information about gathering statistics, see the IBM Tivoli Access Manager Problem Determination Guide.

Examples The following is an example of the output after sending the pdadmin stats list task to the authorization server:
pdadmin> server task ivacld-mogman.admogman.com stats list

d.ras.stats.monitor
d.log.EventPool.queue

Return Codes The following exit status codes can be returned:

0 The command completed successfully.
1 The command failed.

Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexidecimal codes.

See Also server task
server task trace
server task (WebSEAL)
server task add (WebSEAL)
server task trace

Enables the gathering of trace information. This information is stored in a file and used for debugging purposes.

Syntax

```
pdadmin –a admin_id –p password server task server_name–host_name trace list [component]
pdadmin –a admin_id –p password server task server_name–host_name trace set component level [log_agent]
pdadmin –a admin_id –p password server task server_name–host_name trace show [component]
```

Options

- `–a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.
- `–p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `server_name–host_name` Specifies the name of the Tivoli Access Manager server.
  
  A valid Tivoli Access Manager server name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  The `server_name` option is the full expression of the actual machine name and the Tivoli Access Manager component used by this command (such as WebSEAL). The `server_name` option is in the format:

  `access_manager_component-machine_name`

  For example, if the machine name is `cruz` and the Tivoli Access Manager component is WebSEAL, the `server_name` is: `webseald-cruz`

  For a single WebSEAL server, `server_name` is `webseald`. For multiple WebSEAL instances on the same machine, `server_name` is the configured name of the WebSEAL server instance followed by `--webseald`. For example, if the configured name of a WebSEAL instance is `webseal2`, the `server_name` is as follows: `webseal2--webseald`.

  **Note:** The initial WebSEAL server installed on a machine is always named after the machine. For example, if the machine name is `patton`, the `host_name` is `patton`.

- `trace list [component]` Lists all enabled trace components available to gather and report trace information. Specify the `component` option to list a specific component that is enabled (set) for tracing. If the specified component is not enabled, no output is displayed.

- `trace set component level [log_agent]` Sets the trace level and trace message destination for a specific `component` and its subordinates. `level` option values are 1 through 9, with 9 reporting the most detailed level of information in the trace output. The optional `log_agent` specifies a destination for the trace information gathered for the specified component. For more information about event logging, see the [IBM Tivoli Access Manager Base Administrator’s Guide](#).
trace show [component]

Shows the name and level for the specified component. If the component option is not specified, shows the names and levels for all enabled trace components.

Description

For more information about tracing and trace components, see the IBM Tivoli Access Manager Problem Determination Guide.

Examples

The following is an example of showing enabled trace components. Note that WebSEAL–specific components are prefixed with pdweb.

```
pdadmin>server task webseald-<instance>
trace set pdweb.debug 2
pdadmin>server task webseald-<instance>
trace show
pdweb.debug 2
```

Return Codes

The following exit status codes can be returned:

- **0** The command completed successfully.
- **1** The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

server task

"server task stats (WebSEAL)" on page 109

server task (WebSEAL)

"server task add (WebSEAL)" on page 107
**user create**

Creates a Tivoli Access Manager user.

**Syntax**

```bash
pdadmin -a admin_id -p password user create [-gsouser] [-no-password-policy] user_name dn cn sn password [groups]
```

**Options**

- `-a admin_id` Specifies the administrator's ID. If this is the only option that is specified, the user is prompted for the password.
- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.
- `-gsouser` Enables the user's global signon (GSO) capabilities.
- `-no-password-policy` Indicates that password policy is not enforced during the creation of the user account. This non-enforcement has no effect on password policy enforcement after user creation.

**user_name** Specifies the name for the user being created. This name must be unique.

A valid user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of user names: dluca, sec_master, "Mary Jones"

**dn** Specifies the registry identifier assigned to the user being created. The registry identifier must be known before a new user account can be created. The registry identifier must be unique within the user registry.

The format for a distinguished name is similar to: "cn=Mary Jones,ou=Austin,o=Tivoli,c=us"

**cn** Specifies the common name assigned to the user being created. For example: "cn=Mary"

**sn** Specifies the surname of the user being created. For example: "sn=Jones"

**password** Specifies the password set for the new user. Passwords must adhere to the password policies set by the administrator.

**groups** This optional option specifies a list of groups to which the new user is assigned. The format of the group list is a parenthesized list of group names, separated by spaces.

A valid user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of groups: deptD4D, printerusers,
Description

A user is a registered participant of the secure domain. A GSO user is a Tivoli Access Manager user that additionally has the authority to work with Web resources, such as a Web server. When an LDAP-based registry is used, user names are not case sensitive.

The -no-password-policy option to the pdadmin user create command allows the administrator to create the user with an initial password that is not checked by the existing global password policies. If this option is not present in the command, the password provided is checked against the global password policies. In this case, the user create command fails if the password is invalid, and the error message includes information on what conditions were not met.

However, if the administrator applies the pdadmin user modify password command, the -no-password-policy option is not available. Therefore, the modified password is always checked against the global password policy settings.

Examples

1. The following example, entered as one line, create the new user dlucas:
   
   ```
   pdadmin>user create –gsouser dlucas "cn=Diana Lucas,ou=Austin,o=Tivoli,c=US" "Diana Lucas" Lucas mypasswd
   ```

2. The following example, entered as one line, create the new user maryj:
   
   ```
   pdadmin>user create –gsouser maryj "cn=Mary Jones,o=tivoli,c=us" "cn=Mary" "sn=Jones" pwd2pwd2
   ```

To make the user accounts valid, you must use the pdadmin user modify command to set the account-valid flag to yes.

Return Codes

The following exit status codes can be returned:

0  The command completed successfully.

1  The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the IBM Tivoli Access Manager Error Message Reference. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also

user delete
user import
user modify
### user delete

Deletes the specified Tivoli Access Manager user and optionally deletes the user’s information in the user registry.

### Syntax

```
pdadmin –a admin_id –p password user delete [--registry] user_name
```

### Options

- **–a admin_id**  
  Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- **–p password**  
  Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- **–registry**  
  Deletes the user’s information from the user registry. If this option is not specified, the registry user information can be used to create another Tivoli Access Manager user by using the `pdadmin user import` command.

- **user_name**  
  Specifies the name of the account to be deleted. Any resource credentials associated with a user account are automatically removed at the same time the user account is deleted.

  A valid user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of user names: dlucas, sec_master, "Mary Jones"

### Examples

The following example deletes the `dlucas` user:

```
pdadmin>user delete dlucas
```

### Return Codes

The following exit status codes can be returned:

- **0**  
  The command completed successfully.

- **1**  
  The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

  Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

### See Also

- [user create](#)
- [user import](#)
user import

Creates a Tivoli Access Manager user by importing user data that already exists in the user registry.

Syntax

```
user import
```

```
padmin -a admin_id -p password user import [-gsouser] user_name dn [group_name]
```

Options

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- `-gsouser` Specifies that the user has single signon capabilities.

  A valid single signon user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

`user_name` Specifies a unique Tivoli Access Manager user name. This user is created from information that already exists in the user registry. For URAF-based registries, such as Domino and Active Directory, the user name must correspond to a short name already defined for the user being imported from the user registry.

  A valid user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of user names: dluca, sec_master, "Mary Jones"

`dn` Specifies the registry identifier of the user being imported. This identifier must exist in the user registry and must not be associated with another user in the same Tivoli Access Manager secure domain.

  The format for a distinguished name is similar to: "cn=Claude Wright,ou=Austin,o=Tivoli,c=us"

`group_name` Specifies an optional group to which the user is being added.

  A valid group name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  Examples of group names: Credit, Sales, Test-group

Description

Imported user accounts are created invalid by default. To make the user account valid, you must use the `padmin user modify` command to set the `account-valid` flag to `yes`.
Examples
The following example, entered on one line, creates the user mlucas by importing information from the registry user cn=Mike Lucaser,ou=Austin,o=Tivoli,c=US:

```
pdadmin> user import --gsouser mlucas "cn=Mike \nLucaser,ou=Austin,o=Tivoli,c=US"
```

Return Codes
The following exit status codes can be returned:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The command completed successfully.</td>
</tr>
<tr>
<td>1</td>
<td>The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2). Refer to the <a href="#">IBM Tivoli Access Manager Error Message Reference</a>. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.</td>
</tr>
</tbody>
</table>

See Also
- [user create](#)
- [user modify](#)
user list

Lists users by Tivoli Access Manager user name or by registry identifier.

Syntax

```
pdadmin -a admin_id -p password user {list | list-dn} pattern max_return
```

Options

- **-a admin_id**
  Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- **-p password**
  Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- **list pattern max_return**
  Specifies the pattern for the principal name. The pattern can include a mixture of wildcard and string constants, and is case sensitive. For example: `*luca*`

  The `pattern max_return` option specifies the maximum number of entries that are found and returned for a single request. Note that the number returned is also governed by the server configuration, which specifies the maximum number of results that can be returned as part of a search operation).

  The actual maximum returned entries is the minimum number of results between the `pattern max_return` and the configured value on the server, which is taken from the `max-search-size=[0|num_entries]` parameter in the `[ldap]` stanza of the `ldap.conf` configuration file.

  For a discussion of how to limit the number of users returned from the `pdadmin user list` command, see the "IBM Tivoli Access Manager Performance Tuning Guide".

- **list-dn pattern max_return**
  Specifies the pattern for the common name (CN) portion of the user’s registry identifier (excluding the `cn=` component). The pattern can include a mixture of wildcard and string constants, and is case sensitive (for example, `*luca*`). The returned list are users which are defined in the user registry but are not necessarily Tivoli Access Manager users. Users that are not Tivoli Access Manager users can be imported into Tivoli Access Manager by use of the `pdadmin user import` command.

Examples

1. The following example lists the users matching the specified pattern:

   ```
pdadmin> user list *luca* 2
   ```

   Output is similar to the following:

   ```
dlucas
mlucaser
   ```

2. The following example lists the users matching the specified registry identifier:

   ```
pdadmin> user list-dn *luca* 2
   ```

   Output is similar to the following:

   ```
cn=Diana Lucas,ou=Austin,o=Tivoli,c=US
cn=Mike Lucaser,ou=Austin,o=Tivoli,c=US
   ```
Return Codes
The following exit status codes can be returned:

0  The command completed successfully.

1  The command failed. When a command fails, the `pdadmin` command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the *IBM Tivoli Access Manager Error Message Reference* for a list of Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
\[user show\]
user modify
Changes various user account attributes.

Syntax

pdadmin –a admin_id –p password user modify user_name account-valid {yes|no}

pdadmin –a admin_id –p password user modify user_name description description

pdadmin –a admin_id –p password user modify user_name gsouser {yes|no}

pdadmin –a admin_id –p password user modify user_name password password

pdadmin –a admin_id –p password user modify user_name password-valid {yes|no}

Options

– a admin_id
Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

–p password
Specifies the password for the admin_id user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the admin_id is not specified.

account-valid {yes|no}
Enables or disables the specified user account. A user cannot log in with a disabled account.

description description
Modifies the user description.

A valid description is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

If the description contains a space, ensure that you enclose the description in double quotation marks. You can specify an empty string ("") to clear an existing description.

Example of description: "Diana Lucas, Credit Dept HCUS"

gsouser {yes|no}
Enables or disables the single signon capabilities of a user. Valid values are yes and no.

password password
Modifies the user password. The new password must comply with password policies in effect.

password-valid {yes|no}
Validates or invalidates the password for the specified user account. Valid values are yes and no. If the value is no, the password will appear to be expired and the user will be unable to log in using the password until an administrator sets the valid state to yes. Or, the user can authenticate using another method, such as using a certificate.

Another reason a password might expire is because the maximum password age has been exceeded. If you check and find that password-valid is currently set to yes, then try changing the value for the pdadmin policy set max-password-age parameter. Only an administrator or a user that has the proper authority can set the max-password-age policy on a user account. A user cannot set this
policy on his own account. This policy sets the maximum time, in days, that a password will be valid. Time is relative to the last time the password was changed.

When you change the value for password-valid or reset pdadmin policy set max-password-age, the user’s password does not have to be changed.

If you reset a user’s password, the password-valid parameter automatically switches to back to yes, and the max-password-age parameter resets the age to expire. For example, if the maximum password age is set to 30 days, another 30 days begins from the time you reset the user’s password.

[user_name] Specifies the name of the account to be modified.

A valid user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of user names: dlucas, sec_master, "Mary Jones"

1. The following example enables the specified user account:
   pdadmin>user modify dlucas account-valid yes

2. The following example modifies the description of a user account:
   pdadmin>user modify dlucas description "Diana Lucas, Credit Dept HCUS"

3. The following example disables the user’s single signon capabilities.
   pdadmin>user modify dlucas gsouser no

4. The following example changes the password for a user account:
   pdadmin>user modify dlucas password newpasswd

Return Codes
The following exit status codes can be returned:

0 The command completed successfully.
1 The command failed. When a command fails, the pdadmin command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference]. This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

See Also
user create
user import
user show
Displays the properties of the specified user.

Syntax

```
pdadmin -a admin_id -p password user show user_name

pdadmin -a admin_id -p password user show-dn dn

pdadmin -a admin_id -p password user show-groups user_name
```

Options

- `-a admin_id` Specifies the administrator’s ID. If this is the only option that is specified, the user is prompted for the password.

- `-p password` Specifies the password for the `admin_id` user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the `admin_id` is not specified.

- `user_name` Specifies the name of the user to display.

  A valid user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of user names: dLucas, sec_master, "Mary Jones"

- `show-dn dn` Displays the user specified by the user’s identifier in the user registry. The returned user is defined in the user registry, but it is not necessarily a Tivoli Access Manager user. Users that are not Tivoli Access Manager users can be imported into Tivoli Access Manager by use of the `pdadmin user import` command.

  The format for a distinguished name is similar to: "cn=Claude Wright, ou=Austin, o=Tivoli, c=us"

- `show-groups user_name` Displays the groups in which the specified user is a member.

  A valid user name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Examples of user names: dLucas, sec_master, "Mary Jones"

Examples

1. The following example displays the user account information for `testuser`:

   ```
   pdadmin> user show testuser
   ```

   Output is similar to the following:

   ```
   Login ID: testuser
   LDAP DN: cn=testuser,o=tivoli,c=us
   LDAP CN: test
   LDAP SN: test
   Description: a test user
   ```
Is SecUser: yes  
Is GSO user: no  
Account valid: no  
Password valid: yes

2. The following example displays the groups of which the specified user is a member:

   pdadmin>user show-groups dlucas

   Output is similar to the following:
   sales
   credit
   engineering

3. The following example provides additional information about the user when specifying the registry identifier:

   pdadmin>user show-dn "cn=Diana Lucas,ou=Austin,o=Tivoli,c=US"

   Output is similar to the following:
   Login ID: dlucas
   LDAP dn: cn=Diana Lucas,ou=Austin,o=Tivoli
   Inc,c=US
   LDAP cn: Diana Lucas
   LDAP sn: Lucas
   Description: Diana Lucas, Credit Dept HCUS
   IS SecUser: true
   IS GSO user: false
   Account valid: true
   Password valid: true
   Authentication mechanism: Default:LDAP

**Return Codes**

The following exit status codes can be returned:

0  The command completed successfully.

1  The command failed. When a command fails, the **pdadmin** command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

Refer to the [IBM Tivoli Access Manager Error Message Reference](#). This reference provides a list of the Tivoli Access Manager error messages by decimal or hexadecimal codes.

**See Also**

[User list](#)
Chapter 2. Tivoli Access Manager utilities

In addition to the `pdadmin` command utility, Tivoli Access Manager provides the following utilities for your use.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>basslcfg -chgpwd</code></td>
<td>Changes the key database password.</td>
</tr>
<tr>
<td><code>basslcfg -config</code></td>
<td>Configures the Tivoli Access Manager runtime to allow the <code>pdadmin</code> and <code>svrsslcfg</code> utilities to communicate with the Tivoli Access Manager policy server. Also creates a new key and stash file.</td>
</tr>
<tr>
<td><code>basslcfg -getcacert</code></td>
<td>Downloads the root CA certificate to a file.</td>
</tr>
<tr>
<td><code>basslcfg -modify</code></td>
<td>Modifies the Tivoli Access Manager policy server configuration.</td>
</tr>
<tr>
<td><code>basslcfg -ping</code></td>
<td>Pings a Tivoli Access Manager server.</td>
</tr>
<tr>
<td><code>ezinstall_filename</code></td>
<td>Uses scripts or batch file to set up complete Tivoli Access Manager systems in the secure domain.</td>
</tr>
<tr>
<td><code>install_pdrte</code></td>
<td>Installs a Tivoli Access Manager runtime system using an InstallShield wizard.</td>
</tr>
<tr>
<td><code>mgrsslcfg -chgcert</code></td>
<td>Renews the SSL certificate of the manager.</td>
</tr>
<tr>
<td><code>mgrsslcfg -chgpwd</code></td>
<td>Changes the key database password.</td>
</tr>
<tr>
<td><code>mgrsslcfg -config</code></td>
<td>Performs full configuration, creating new key and stash files and generating new certificates for the Tivoli Access Manager policy server.</td>
</tr>
<tr>
<td><code>mgrsslcfg -modify</code></td>
<td>Modifies the current configuration.</td>
</tr>
<tr>
<td><code>pdbackup</code></td>
<td>Backs up, restores, and extracts Tivoli Access Manager data.</td>
</tr>
<tr>
<td><code>pdconfig</code></td>
<td>Configures and unconfigures Tivoli Access Manager components except the Tivoli Access Manager Java runtime environment component.</td>
</tr>
<tr>
<td><code>pdrtecfg</code></td>
<td>Configures the Tivoli Access Manager Java runtime environment.</td>
</tr>
<tr>
<td><code>pd_start</code></td>
<td>Stops, starts, and restarts servers on UNIX systems. Also displays server status.</td>
</tr>
<tr>
<td><code>pdversion</code></td>
<td>Lists the current version of Tivoli Access Manager components installed on the system.</td>
</tr>
<tr>
<td><code>svrsslcfg</code></td>
<td>Configures aznAPI applications to use an SSL connection for communicating with the Tivoli Access Manager policy server.</td>
</tr>
<tr>
<td><code>svrsslcfg -add_replica</code></td>
<td>Adds a database replica.</td>
</tr>
<tr>
<td><code>svrsslcfg -chg_replica</code></td>
<td>Changes a database replica.</td>
</tr>
<tr>
<td><code>svrsslcfg -chgcert</code></td>
<td>Renews the application server’s SSL certificate.</td>
</tr>
<tr>
<td><code>svrsslcfg -chgpport</code></td>
<td>Changes the listening port number.</td>
</tr>
<tr>
<td><code>svrsslcfg -chgpwd</code></td>
<td>Changes the keyring file password.</td>
</tr>
<tr>
<td><code>svrsslcfg -config</code></td>
<td>Performs full configuration of an application server.</td>
</tr>
<tr>
<td><code>svrsslcfg -modify</code></td>
<td>Modifies the current configuration.</td>
</tr>
<tr>
<td><code>svrsslcfg -rmv_replica</code></td>
<td>Removes a replica configuration.</td>
</tr>
<tr>
<td><code>svrsslcfg -unconfig</code></td>
<td>Unconfigures the application server.</td>
</tr>
</tbody>
</table>
bassslcfg –chgpwd

Changes the key database password. A new random password is generated and saved in the stash file.

Syntax
bassslcfg –chgpwd –e pwd_life

Options
–e pwd_life   Sets the keyring file password expiration time in days. You can specify a pwd_life value from 1 to 7200 (days). To use the currently configured value, specify 0.

Availability
This command is located in the following default installation directories:
• On UNIX systems:
  /opt/PolicyDirector/sbin/
• On Windows systems:
  c:\Program Files\Tivoli\Policy Director\sbin\n
Return Codes
The following exit status codes can be returned:
0   The command completed successfully.
1   The command failed.
bassslcfg –config

Configures the Tivoli Access Manager runtime so as to allow the pdadmin and svrsslcfg utilities to communicate with the Tivoli Access Manager policy server. Also creates a new key and stash file.

Syntax
bassslcfg –config –c cert_file –h host_name [–p server_port] [–e pwd_life] [–t ssl_timeout]

Options

–c cert_file Specifies the name of the policy server base64-encoded, self-signed certificate.

–h host_name Specifies the TCP host name of the policy server.
   Valid values for host_name include any valid IP host name.

   Examples:
   host = libra
   host = libra.dallas.ibm.com

–p server_port Specifies the listening port number of the policy server. The default value is 7135.
   For server_port, use any valid port number. A valid port number is any positive number that is allowed by TCP/IP and that is not currently being used by another application. It is recommended that you use the default port number value of 7135, or else use a port number over 1000 currently not being used.

–e pwd_life Sets the keyring file password expiration time in days. You can specify a pwd_life value from 1 to 7299 (days). The default value is 7299.

–t ssl_timeout Specifies the SSL session timeout in seconds. You can specify an ssl_timeout value from 1 to 86400 (seconds). The default value is 7200.

Availability
This command is located in the following default installation directories:

- On UNIX systems:
  /opt/PolicyDirector/sbin/
- On Windows systems:
  c:\Program Files\Tivoli\Policy Director\sbin\

Return Codes
The following exit status codes can be returned:

0   The command completed successfully.
1   The command failed.
**bassslcfg –getcacert**

Downloads the root CA certificate to a file.

**Syntax**

```
bassslcfg –getcacert –c cert_file –h host_name [–p server_port]
```

**Options**

- `-c cert_file` Specifies the name of the policy server base-64 encoded, self-signed certificate.
- `-h host_name` Specifies the TCP host name of the policy server. Valid values for `host_name` include any valid IP host name.
  
  Examples:
  
  host = libra
  host = libra.dallas.ibm.com

- `-p server_port` Specifies the listening port number of the policy server. The default value is 7135.

**Availability**

This command is located in the following default installation directories:

- On UNIX systems:
  
  `/opt/PolicyDirector/sbin/`

- On Windows systems:
  
  `c:\Program Files\Tivoli\Policy Director\sbin\`

**Return Codes**

The following exit status codes can be returned:

- `0` The command completed successfully.
- `1` The command failed.
bassslcfg –modify

Modifies the Tivoli Access Manager policy server configuration.

Syntax

bassslcfg –modify [–h host_name] [–e pwd_life] [–p server_port] [–t ssl_timeout]

Options

–h host_name

Specifies the TCP host name of the policy server.

Valid values for host_name include any valid IP host name.

Examples:

host = libra
host = libra.dallas.ibm.com

–e pwd_life

Sets the keyring file password expiration time in days. You can specify a pwd_life value from 1 to 7200 (days).

–p server_port

Specifies the listening port number of the policy server. The default value is 7135.

For server_port, use any valid port number. A valid port number is any positive number that is allowed by TCP/IP and that is not currently being used by another application. It is recommended that you use the default port number value of 7135, or else use a port number over 1000 currently not being used.

–t ssl_timeout

Specifies the SSL session timeout in seconds. You can specify an ssl_timeout value from 1 to 86400 (seconds).

Availability

This command is located in the following default installation directories:

• On UNIX systems:
  /opt/PolicyDirector/sbin/

• On Windows systems:
  c:\Program Files\Tivoli\Policy Director\sbin\

Return Codes

The following exit status codes can be returned:

0   The command completed successfully.
1   The command failed.
**bassslcfg -ping**

Pings a Tivoli Access Manager server.

**Syntax**

*bassslcfg -ping -h* *host_name* [*-p* *server_port*]

**Options**

*–h host_name* Specifies the TCP host name of the policy server.

Valid values for *host_name* include any valid IP host name.

Examples:

- *host = libra*
- *host = libra.dallas.ibm.com*

*–p server_port* Specifies the listening port number of the Tivoli Access Manager server that you want to ping. The default value is 7135.

**Availability**

This command is located in the following default installation directories:

- On UNIX systems:
  `/opt/PolicyDirector/sbin/`
- On Windows systems:
  `c:\Program Files\Tivoli\Policy Director\sbin\`

**Return Codes**

The following exit status codes can be returned:

- 0 The command completed successfully.
- 1 The command failed.
ezinstall_filename

Expedites the installation and configuration of a secure domain using an LDAP registry. The easy-installation files are also useful if you want to add a Tivoli Access Manager component or set up a system in an existing domain.

For detailed information, including step-by-step scenarios, see the IBM Tivoli Access Manager Base Installation Guide. Ensure that you are familiar with the configuration options of the ezinstall_filename files. Before running, ensure that the following easy installation files are supported on your platform:

Syntax

```
ezinstall_ldap_server [response_file]
```

```
ezinstall_pdacld [response_file]
```

```
ezinstall_pdauthadk [response_file]
```

```
ezinstall_pdmgr [response_file]
```

```
ezinstall_pdwpm [response_file]
```

Options

**ezinstall_ldap_server**

Sets up an IBM Directory server system with the following software packages:

- IBM DB2
- IBM Global Security Toolkit
- IBM HTTP Server
- IBM Directory client
- IBM Directory server

**ezinstall_pdacld**

Sets up an authorization server system with the following software packages:

- IBM Global Security Toolkit
- IBM Directory client
- Tivoli Access Manager runtime
- Tivoli Access Manager authorization server

**ezinstall_pdauthadk**

Sets up a Tivoli Access Manager development system with the following software packages:

- IBM Global Security Toolkit
- IBM Directory client
- Tivoli Access Manager runtime
- Tivoli Access Manager Application Development Kit

**ezinstall_pdmgr**

Sets up the Tivoli Access Manager policy server system with the following software packages:

- IBM Global Security Toolkit
- IBM Directory client
- Tivoli Access Manager runtime
- Tivoli Access Manager policy server
ezinstall_pdwpm

Sets up the Web Portal Manager interface with the following software packages:
- IBM Global Security Toolkit
- IBM Directory client
- Tivoli Access Manager runtime
- IBM WebSphere Application Server, Advanced Single Server 4.0, and Fix Pack 3
- Tivoli Access Manager Web Portal Manager
- Tivoli Access Manager Java runtime environment

response_file

Specifies a response file to perform a silent, unattended installation of Tivoli Access Manager components.

On UNIX systems, the response file is named based on the package that you installed and configured. For example, if you run the ezinstall_ldap_server script, the response file that is generated is named ezinstall_ldap_server.rsp. Response files for each package that you run are stored in the /var/tmp directory.

On Windows systems, easy installation generates a response file named ezinstallrsp. This response file resides in the temporary directory that is the value specified by the %TEMP% variable. For example, if you run the ezinstall_ldap_server.bat file, the response file that is generated is named %TEMP%\ezinstall.rsp.

**Note:** For more information about response files, see the IBM Tivoli Access Manager Base Installation Guide.

Comments

- If you plan to configure Active Directory or Domino as your registry, you cannot use easy installation. In addition, easy installation is not supported on the Linux for zSeries platform.
- You cannot use the ezinstall_ldap_server script if an existing version of IBM Directory server is installed.
- The ezinstall_pdmgr script is supported on AIX, Solaris, and Windows systems only. You cannot use the ezinstall_pdmgr script if an existing version of the Tivoli Access Manager policy server is installed.

Return Codes

The following exit status codes can be returned:

0 The command completed successfully.
1 The command failed.

See Also

install_pdrte
install_pdrte

Sets up a Tivoli Access Manager runtime system. All prerequisite products and Tivoli Access Manager components are installed and configured except for a platform-specific JRE, which must be installed manually.

Before running install_pdrte, ensure that the install_pdrte utility is supported on your platform and that you are familiar with its configuration options. For detailed information, including step-by-step scenarios, see the IBM Tivoli Access Manager Base Installation Guide.

Syntax
install_pdrte [-options {response_file}]

Options
install_pdrte Sets up a Tivoli Access Manager runtime system with the following software packages:

- IBM Global Security Toolkit
- IBM Directory client
- Access Manager Runtime

Comments
- This utility is only supported when using an LDAP-based registry.
- To create a Tivoli Access Manager runtime response file, you must copy a template provided on the Tivoli Access Manager Base CD to your hard drive and edit its values.

Return Codes
The following exit status codes can be returned:

0 The command completed successfully.
1 The command failed.

See Also
ezinstall_filename
**mgrsslcfg –chgcert**

Renews the SSL certificate of the manager. A new public-private key pair and certificate are created and stored in the key database.

**Syntax**

```
mgrsslcfg –chgcert –l cert_life
```

**Options**

`–l cert_life`  Sets the certificate expiration time in days. You can specify a `cert_life` value from 1 to 7299 (days). To use the currently configured value, specify 0.

**Comments**

Stop the policy server before running this command.

**Availability**

This command is located in the following default installation directories:

- On UNIX systems:
  `/opt/PolicyDirector/sbin/
- On Windows systems:
  `c:\Program Files\Tivoli\Policy Director\sbin`

**Return Codes**

The following exit status codes can be returned:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The command completed successfully.</td>
</tr>
<tr>
<td>1</td>
<td>The command failed.</td>
</tr>
</tbody>
</table>
mgrsslcfg –chgpwd
Changes the key database password. A new random password is generated and saved in the stash file.

Syntax
mgrsslcfg –chgpwd –e pwd_life

Options
–e pwd_life Sets the keyring file password expiration time in days. You can specify a pwd_life value from 1 to 7200 (days). To use the currently configured value, specify 0.

Comments
Stop the policy server before running this command.

Availability
This command is located in the following default installation directories:
• On UNIX systems:
  /opt/PolicyDirector/sbin/
• On Windows systems:
  c:\Program Files\Tivoli\Policy Director\sbin\n
Return Codes
The following exit status codes can be returned:
0 The command completed successfully.
1 The command failed.
mgrsslcfg –config

Performs full configuration, creating new key and stash files and generating new certificates for the Tivoli Access Manager policy server.

Syntax

mgrsslcfg –config [–e pwd_life] [–l cert_life] [–t ssl_timeout] [–D {yes|no}]

Options

–e pwd_life   Sets the keyring file password expiration time in days. The pwd_life value is 1 to 7200 (days). If not specified a default value of 183 is used.
–l cert_life  Sets the certificate expiration time in days. You can specify a cert_life value from 1 to (days). The default value is 365.
–t ssl_timeout  Specifies the SSL session timeout in seconds. You can specify an ssl_timeout value from 1 to 86400 (seconds). The default value is 7200.
–D {yes|no}  Specifies whether hosts can download the secure domain’s CA certificate. If you specify no, you must copy or transfer the pdcacert.b64 file to subsequent hosts in order to configure a Tivoli Access Manager runtime. The default value is no.

Availability

This command is located in the following default installation directories:
- On UNIX systems:
  /opt/PolicyDirector/sbin/
- On Windows systems:
  c:\Program Files\Tivoli\Policy Director\sbin\n
Return Codes

The following exit status codes can be returned:
0   The command completed successfully.
1   The command failed.
**mgrsslcfg –modify**

Modifies the current configuration.

**Syntax**

```shell
mgrsslcfg –modify [–e pwd_life] [–l cert_life] [–t ssl_timeout] [–D {yes|no}]
```

**Options**

- **–e pwd_life** Sets the keyring file password expiration time in days. The `pwd_life` value is 1 to 7200 (days).
- **–l cert_life** Sets the certificate expiration time in days. The range of values is 1 to 7299.
- **–t ssl_timeout** Specifies the SSL session timeout in seconds. The `ssl_timeout` value must be in the range 1–86400.
- **–D {yes|no}** Enables downloading of the secure domain’s CA certificate. If `no` is specified, you must manually copy the `pdcacert.b64` file to subsequent hosts before configuring the Tivoli Access Manager runtime component.

**Availability**

This command is located in the following default installation directories:

- On UNIX systems:
  ```shell
  /opt/PolicyDirector/sbin/
  ```
- On Windows systems:
  ```shell
  c:\Program Files\Tivoli\Policy Director\sbin\n  ```

**Return Codes**

The following exit status codes can be returned:

- **0** The command completed successfully.
- **1** The command failed.
**pdbackup**

Backs up, restores, and extracts Tivoli Access Manager data.

**Syntax**

```
pdbackup -action backup -list path_to_backup_list [-path path][-file filename] [-usage ][-?]  
pdbackup -action restore -file filename [-path path][-usage][-?]  
pdbackup -action extract -file filename -path path[-usage][-?]  
```

**Options**

Note that you can shorten an option name, but the abbreviation must be unambiguous. For example, you can type `a` for `action`. However, values for options cannot be shortened.

- `-action [backup | restore | extract]` - Specifies to backup, restore, or extract data.
- `-list path_to_backup_list` - Specifies the fully qualified path to the backup list file—an ASCII file containing various stanzas. This option is required when using the `-a backup` option.
- `-path path` - Specifies one of the following:
  - If specified with the `-a backup` option, specifies the path where you want backed up files stored. If you do not specify a path when using the `-a backup` option, the default path is one of the following:
    - On UNIX systems, the default path is as follows:
      `/var/PolicyDirector/pdbackup/`
    - On Windows systems, the default path is as follows:
      `runtime_dir\pdbackup\`
      where `runtime_dir` specifies the directory where the Tivoli Access Manager runtime is installed.
  - If specified with the `-a restore` option on UNIX systems only, indicates to restore archived files in the specified `path`. By default, the restore path is on the directory used when backing up data. On Windows systems, the restore process does not support the `-p` option.
  - If specified with the `-a extract` option, specifies the directory name where you want extracted files stored. There is no default path. The `-p` option is required when using the `-a extract` option.
- `-file filename` - Specifies one of the following:
  - If specified with the `-a backup` option, specifies a file name other than the `list_date.time [.tar|.dir]` default file name.
  - If specified with the `-a restore` option, specifies the name and fully qualified path of the archive file to restore. There is no default path. This option is required when using the `-a restore` option.
  - If specified with the `-a extract` option, specifies the name and fully qualified path of the archive file to extract. There is no default path. This option is required when using the `-a extract` option.

A valid file name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period
Comments
Use this command to back up and restore Tivoli Access Manager data.

Archived files are stored in one of the following ways:

- On UNIX systems, the archive is stored as a single .tar file in the /var/PolicyDirector/pdbackup default directory. The default file name is as follows:

  list_date.time.tar

  where list is the name specified by the –list option and date.time is the current date and timestamp of the archived file.

- On Windows systems, the archive is stored as a directory tree in the \runtime_environment_path\pdbackup default directory.

Files are restored in one of the following ways:

- On UNIX systems, archived files are restored to the root directory unless you specify the –path option, which enables you to restore files to a specific directory tree.

- On Windows systems, archived files are restored to their original directory. There is no –path option available.

You also can use this command during the upgrade process to extract files in a single directory (without a directory tree structure). Note that Windows registry keys are not updated with the –a extract option.

Availability
This command is located in the following default installation directories:

- On UNIX systems:
  /opt/PolicyDirector/bin/

- On Windows systems:
  c:\Program Files\Tivoli\Policy Director\bin\

UNIX Examples
1. The following example performs a back up with default values:

   pdbackup -a backup -l /opt/PolicyDirector/etc/pdbackup.lst

   This results in a file named pdbackup.lst_date.time.tar, located in the
   /var/PolicyDirector/pdbackup directory.

2. The following example performs a back up, creating the default archive file in the /var/backup directory:

   pdbackup -a backup -l /opt/PolicyDirector/etc/pdbackup.lst -p /var/backup

   This results in a file named pdbackup.lst_date.time.tar, located in the /var/backup directory.
3. The following example performs a back up, creating a file named pdarchive.tar in the following default path:

```
pdbackup -a backup -list /opt/PolicyDirector/etc/pdbackup.1st -f pdarchive
```

The default archive extension (.tar) is appended to the pdarchive file name. This file is stored in the /var/PolicyDirector/pdbackup directory.

4. The following example restores the archive file in the default location:

```
pdbackup -a restore -f pdbackup.1st_29June2002.07_24.tar
```

5. The following example restores the archive file from the /var/pdback directory:

```
pdbackup -a restore -f /var/pdback/pdbackup.1st_29Jun2002.07_25.tar
```

6. The following example restores the archive file from the /var/pdback directory to a directory named /pdtest:

```
pdbackup -a restore -p pdtest -f /var/pdback/
pdbackup.1st_29Jun2002.07_25.tar
```

7. The following example extracts the contents of an archive file to a directory named e:/pdextract. The 

```
pdbackup -a extract -p e:\pdextract -f c:\pdbackup\pdbackup.1st_29Jun2002.07_25.tar
```

If the pdextract directory does not exist, it is created. Note that all files in the archive file are copied to this single directory. No subdirectories are created.

**Windows Examples**

1. The following example performs a standard backup with default values:

```
pdbackup -a backup -l base_dir\etc\pdbackup.1st
```

This results in a file named pdbackup.1st_date.time.dir, located in the base_dir\pdbackup directory.

2. The following example performs a back up by using the default archive file name and by storing the file in the c:\pdback directory:

```
pdbackup -a backup -l base_dir\etc\pdbackup.1st -path c:\pdback
```

3. The following example performs a back up using the default path with a file named pdarchive.dir:

```
pdbackup -a backup -l base_dir\etc\pdbackup.1st -f pdarchive
```

The default archive extension (.dir) is applied to the pdarchive file name. The file is stored in the base_dir\pdbackup directory.

4. The following example performs a back up to the pdback directory on the F drive:

```
pdbackup -a backup -l pdbackup.1st -p f:\pdback
```

5. The following example restores the archive file from the default directory:

```
pdbackup -a restore -f base_dir\etc\pdbackup.1st_29Jun2002.07_24.dir
```

6. The following example restores files from the c:\pdbackup directory:

```
pdbackup -a restore -f h:\pdbackup\pdbackup.1st_29Jun2002.07_25.dir
```

7. The following example extracts the contents of an archive to the e:\pdextract directory from the c:\pdback directory:

```
pdbackup -a extract -p e:\pdextract
          -f c:\pdback\pdbackup.1st_29Jun2002.07_25.dir
```

**Return Codes**

The following exit status codes can be returned:

0  The command completed successfully.
1  The command failed.
pdconfig

Presents the user with an interactive menu to configure and unconfigure Tivoli Access Manager components with the exception of the Tivoli Access Manager Java runtime component, which uses pdjrtecfg. See the IBM Tivoli Access Manager Base Installation Guide for instructions on how to use this utility.

Syntax

pdconfig

Options

None

Availability

This command is located in the following default installation directories:

- On UNIX systems:
  /opt/PolicyDirector/bin/
- On Windows systems:
  c:\Program Files\Tivoli\Policy Director\bin\

Return Codes

The following exit status codes can be returned:

0  The command completed successfully.
1  The command failed.
**pdjrtecfg**

Configures the Tivoli Access Manager Java runtime environment. The Tivoli Access Manager Java runtime environment enables Java applications to manage and use Tivoli Access Manager security.

**Syntax**

```plaintext
pdjrtecfg –action config [–java_home { jre_path}][–rspfile filename]

pdjrtecfg –action unconfig –java_home { all | jre_path} [–remove_common_jars {yes | no}][–rspfile filename]

pdjrtecfg –action config –host mgmt_host –port mgmt_port [–java_home jre_home] [–domain domain_name] [–rspfile filename]

pdjrtecfg –action unconfig –java_home { jre_home | all} [–remove_common_jars {yes | no}][–rspfile filename]

pdjrtecfg [–help [options]] [–operations] [–usage]
```

**Options**

- **–action {config | unconfig}** Specifies to configure or unconfigure the Tivoli Access Manager Java runtime environment.

- **–help** Prints descriptions of the valid command line options.

- **–java_home jre_path** Specifies the fully-qualified path to the Java runtime environment (such as the directory ending in JRE). For example: `c:\Program Files\IBM\JAVA13\JRE`

  During unconfiguration (–action unconfig), you can use the all suboption, which unconfigures all configured JREs. During configuration (–action config), the jre_path variable is not required. If a path is not specified, the current JRE (specified in the path) is used.

- **–remove_common_jars {yes | no}**

  During unconfiguration only, specifies to delete (yes) or not to delete (no) other IBM related jars, such as logging and security jar files. If this option is not specified, the IBM-related jars are not deleted during unconfiguration.

- **–operations** Prints out all the valid command line options.

- **–rspfile filename** Specifies to use a response file named filename. There is not a default response file name.

  A valid file name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  **Note:** For Windows, file names cannot have these characters: a forward slash (/), a backward slash (/), a colon (:), a question mark (?), double quotation marks, and an asterisk (*).

- **–usage** Prints out the usage information.

- **–?** Prints the usage information.
Comments
This command copies Tivoli Access Manager Java libraries to a library extensions directory that exists for a Java runtime that has already been installed on the system.

Using this command does not overwrite Jar files that already exist in the jre_home\lib\ext directory, except the PD.jar file, which is overwritten if the file exists.

You can install more than one Java runtime on a given machine. The pdjrtecfg command can be used to configure the Tivoli Access Manager Java runtime independently to each of the JREs.

Make sure that you use the pdjrtecfg utility and not the pdjrtecfg Java class directly.

Examples
1. The following example configures the Tivoli Access Manager Java runtime environment:
   pdjrtecfg -action config -java_home E:\apps\IBM\Java131\jre
2. The following example unconfigures the Tivoli Access Manager Java runtime environment:
   pdjrtecfg -action unconfig -java_home E:\apps\IBM\Java131\jre
   -remove_common_jars yes

Availability
This command is located in the following default installation directories:
- On UNIX systems:
  /opt/PolicyDirector/sbin/
- On Windows systems:
  c:\Program Files\Tivoli\Policy Director\sbin\n
Return Codes
The following exit status codes can be returned:
0  The command completed successfully.
1  The command failed.
pd_start

Stops, starts, and restarts servers on UNIX systems. Also displays server status.

Note: On Windows systems use the Services folder.

Syntax

pd_start {restart | start | status | stop}

Options

start
Starts all Tivoli Access Manager servers not currently running on the local system.

stop
Stops all Tivoli Access Manager servers not currently running on the local system.

restart
Restarts all configured all Tivoli Access Manager servers.

status
Displays the state of all configured Tivoli Access Manager servers (running or stopped).

Comments

Server processes are normally enabled and disabled through automated scripts that run at system startup and shutdown. In a UNIX environment, you can also use the pd_start script to manually start and stop the server processes. This technique is useful when you need to customize an installation or when you need to perform troubleshooting tasks.

You can only use pd_start to run scripts on the local machine.

Availability

This command is located in the following default installation directory on UNIX systems:
/opt/PolicyDirector/bin/

Return Codes

The following exit status codes can be returned:

0 The command completed successfully.
1 The command failed.
pdversion

Lists the current version of Tivoli Access Manager components installed on the system.

Syntax

\texttt{pdversion} \[–key \texttt{key1, key2}... \] \[–separator <\texttt{delimiter_character}>\]

Options

\texttt{key} \quad \text{Specifies the component for which the current version will be presented. Possible values are as follows:}
\begin{itemize}
  \item \texttt{pdrt}
  \item \texttt{pdmgr}
  \item \texttt{pdwpm}
  \item \texttt{pdacld}
  \item \texttt{pdjrte}
  \item \texttt{pdauthadk}
\end{itemize}

Examples

The following example lists Tivoli Access Manager components and indicates the version number for any components installed of the current system:

\begin{verbatim}
pdversion
IBM Tivoli Access Manager Runtime 4.1.0.0
IBM Tivoli Access Manager Policy Server 4.1.0.0
IBM Tivoli Access Manager Web Portal Manager Not Installed
IBM Tivoli Access Manager Application Developer Kit 4.1.0.0
IBM Tivoli Access Manager Authorization Server 4.1.0.0
IBM Tivoli Access Manager Java Runtime Environment Not Installed
\end{verbatim}

Availability

This command is located in the following default installation directories:
\begin{itemize}
  \item On UNIX systems:
    \texttt{/opt/PolicyDirector/sbin/}
  \item On Windows systems:
    \texttt{c:\Program Files\Tivoli\Policy Director\sbin/}
\end{itemize}

Return Codes

The following exit status codes can be returned:
\begin{itemize}
  \item 0 \quad \text{The command completed successfully.}
  \item 1 \quad \text{The command failed.}
\end{itemize}
svrsslcfg –add_replica

Adds an authorization server replica to an application server’s configuration. An application server can contact a replica server to perform authorization decisions.

Syntax

svrsslcfg –add_replica –f cfg_file –h host_name [–p server_port] [–k replica_rank]

Options

–f cfg_file  Specifies the configuration file path and name.  
A valid file name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Note: For Windows, file names cannot have these characters: a forward slash (/), a backward slash (\), a colon(:), a question mark (?), double quotation marks, and an asterisk (*).

For example:

UNIX /opt/PolicyDirector/etc/activedir.conf
Windows C:\Program Files\Tivoli\Policy Director\etc\activedir.conf

–h host_name  Specifies the TCP host name of an authorization server replica.

Valid values for host_name include any valid IP host name.

Examples:

host = libra
host = libra.dallas.ibm.com

–p server_port  Specifies the port number on which the replica server listens for requests. The default value is 7136.

–k replica_rank  Specifies the replica order of preference among other replicas. The default value is 10. Replica servers with higher rankings are used preferentially. For example, an application server contacts a replica server with a ranking of 10 before contacting a replica server with a ranking of 9.

Availability

This command is located in the following default installation directories:

• On UNIX systems:
  /opt/PolicyDirector/bin/
• On Windows systems:
  C:\Program Files\Tivoli\Policy Director\bin

Return Codes

The following exit status codes can be returned:

0  The command completed successfully.
1  The command failed.
svrsslcfg –chg_replica

Changes replica server attributes. The replica host name is used to identify the replica server and cannot be changed by this action.

Syntax

svrsslcfg –chg_replica –f cfg_file –h host_name [–p server_port] [–k replica_rank]

Options

–f cfg_file

Specifies the configuration file path and name.

A valid file name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Note: For Windows, file names cannot have these characters: a forward slash (/), a backward slash (/), a colon(:), a question mark (?), double quotation marks, and an asterisk (*).

For example:

UNIX
  /opt/PolicyDirector/etc/activedir.conf

Windows
  C:\Program Files\Tivoli\Policy Director\etc\activedir.conf

–h host_name

Specifies the TCP host name of an authorization server replica.

Valid values for host_name include any valid IP host name.

Examples:

  host = libra
  host = libra.dallas.ibm.com

–p server_port

Specifies the port number on which the replica server listens for requests.

For server_port, use any valid port number. A valid port number is any positive number that is allowed by TCP/IP and that is not currently being used by another application. It is recommended that you use the default port number value of 7136, or else use a port number over 1000 currently not being used.

–k replica_rank

Specifies the replica order of preference among other replicas. Replica servers with higher rankings are used preferentially. For example, an application server contacts a replica server with a ranking of 10 before contacting a replica server with a ranking of 9.

Availability

This command is located in the following default installation directories:

- On UNIX systems:
  /opt/PolicyDirector/bin/
- On Windows systems:
  c:\Program Files\Tivoli\Policy Director\bin\
Return Codes
The following exit status codes can be returned:

0  The command completed successfully.
1  The command failed.
svrsslcfg –chgcert
Replaces the SSL certificate for the application server.

Syntax
svrsslcfg –chgcert –f cfg_file –n appl_name [–P admin_pwd] [–A admin_id]

Options
–f cfg_file
   Specifies the configuration file path and name.
   A valid file name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).
   
   Note: For Windows, file names cannot have these characters: a forward slash (/), a backward slash (/), a colon(:), a question mark (?), double quotation marks, and an asterisk (*).
   
   For example:
   UNIX /opt/PolicyDirector/etc/activedir.conf
   Windows C:\Program Files\Tivoli\PolicyDirector\etc\activedir.conf

–n appl_name
   Specifies the name of the application. The name is combined with the host name to create unique names for Tivoli Access Manager objects created for your application. The following names are reserved for Tivoli Access Manager applications: ivacld, secmgrd, ivnet, and ivweb.

–P admin_pwd
   Specifies the Tivoli Access Manager administrator password. If this option is not specified, the password is read from standard input.

–A admin_id
   Specifies the Tivoli Access Manager administrator name. The default is sec_master.
   
   The ID is an alphanumeric, case-insensitive string. The minimum and maximum lengths of the ID, if there are limits, are imposed by the underlying registry. For Active Directory, the maximum length is 256 alphanumeric characters.

Comments
Stop the policy server before running this command.

The certificate replacement process is as follows:

- When an initial request for a certificate is made, a new public/private key pair is generated for the application server along with the certificate request. The certificate request, which contains the application server's new public key, is sent to the Tivoli Access Manager policy server (pdmgrd). The policy server signs the request and sends the newly signed certificate back to the application server. The application server stores the signed certificate in a secure keystore and also stores the application server's new private key.

- When the certificate is renewed, that original request is sent to the Tivoli Access Manager policy server (pdmgrd), which sends back a new certificate. This new certificate has the application server's original public key, but with a new serial number and expiration date. The application server stores this new certificate in a secure keystore along with the original private key because that key is still the mate to
the public key in the certificate. The conditions for renewing the certificate are if the certificate has expired or if the certificate must be changed to adhere to the enterprise security policy.

- When the application server’s private key has been compromised, the application server must be unconfigured and reconfigured so that another new key pair and certificate are generated.

**Availability**

This command is located in the following default installation directories:

- On UNIX systems:
  /opt/PolicyDirector/bin/
- On Windows systems:
  c:\Program Files\Tivoli\Policy Director\bin\

**Return Codes**

The following exit status codes can be returned:

0  The command completed successfully.

1  The command failed.
svrsslcfg –chgport

Changes an application server's listening port number.

Syntax
svrsslcfg –chgport –f cfg_file –r port_number

Options
–f cfg_file
  Specifies the configuration file path and name.
  A valid file name is an alphanumeric string that can be specified using any case. Valid string
  characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward
  slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an
  asterisk (*).

  Note: For Windows, file names cannot have these characters: a forward slash (/), a backward slash
  (\), a colon (:), a question mark (?), double quotation marks, and an asterisk (*).

  For example:
  UNIX  /opt/PolicyDirector/etc/activedir.conf
  Windows C:\Program Files\Tivoli\Policy
           Director\etc\activedir.conf

–r port_number
  Sets the listening port number for the server. A value of 0 may be specified only if the
  [aznapi-admin-services] stanza in the configuration file is empty.

Comments
Stop the Tivoli Access Manager policy server before running this command.

Availability
This command is located in the following default installation directories:
• On UNIX systems:
  /opt/PolicyDirector/bin/
• On Windows systems:
  c:\Program Files\Tivoli\Policy Director\bin

Return Codes
The following exit status codes can be returned:
  0  The command completed successfully.
  1  The command failed.
svrsslcfg –chgpwd
Changes an application server’s keyring file password.

Syntax
svrsslcfg –chgpwd –f cfg_file –e pwd_life

Options
–f cfg_file Specifies the configuration file path and name.
   A valid file name is an alphanumeric string that can be specified using any case. Valid
   string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal
   sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign
   (@), an ampersand (&), and an asterisk (*).

   Note: For Windows, file names cannot have these characters: a forward slash (/), a
   backward slash (\), a colon(:), a question mark (?), double quotation marks, and an
   asterisk (*).

   For example:
   UNIX   /opt/PolicyDirector/etc/activedir.conf
   Windows  C:\Program Files\Tivoli\Policy Director\etc\activedir.conf

–e pwd_life Sets the keyring file password expiration time in days. The pwd_life value is 1 to 7200
   (days). To use the currently configured value, specify 0.

Comments
Stop the Tivoli Access Manager policy server before running this command.

Availability
This command is located in the following default installation directories:
• On UNIX systems:
  /opt/PolicyDirector/bin/
• On Windows systems:
  c:\Program Files\Tivoli\Policy Director\bin\

Return Codes
The following exit status codes can be returned:
0  The command completed successfully.
1  The command failed.
svrsslcfg –config
Performs full configuration of an application server.

Syntax
[–l listening_mode] [–a refresh_mode] [–C cert_file] [–h host_name]

Options
–f cfg_file   Specifies the configuration file path and name.

A valid file name is an alphanumeric string that can be specified using any
case. Valid string characters are the letters a-Z, the numbers 0-9, a period
(.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_),
a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an
asterisk (*).

Note: For Windows, file names cannot have these characters: a forward
slash (/), a backward slash (\), a colon(:), a question mark (?),
double quotation marks, and an asterisk (*).

For example:
UNIX       /opt/PolicyDirector/etc/activedir.conf
Windows  C:\Program Files\Tivoli\Policy
           Director\etc\activedir.conf

–d kdb_dir   Specifies the directory that is to contain the keyring database files for the
server.

For example:
UNIX       /opt/PolicyDirector/keytab/ivmgrd.kbd
Windows  C:\Program Files\Tivoli\Policy
           Director\keytab\ivmgrd.kbd

–n appl_name   Specifies the name of the application. The name is combined with the host
name to create unique names for Tivoli Access Manager objects created
for your application. The following names are reserved for Tivoli Access
Manager applications: ivacld, secmgrd, ivnet, and ivweb.

–s server_type   Specifies the type of server being configured. The value must be either
local or remote.

–r port_number   Sets the listening port number for the server. This is a required option. A
value of 0 may be specified only if the [aznapi-admin-services] stanza in
the configuration file is empty.

–P admin_pwd   Specifies the Tivoli Access Manager administrator password. This is a
required option. If this option is not specified, the password is read from
standard input.

–S server_pwd   Specifies the server’s password. This option is required. However, you can
request that a password be created by the system by specifying a dash
(–) for the password. If this option is used, the configuration file is updated
with the password created by the system. If the user registry type is LDAP
and a password is specified, it is saved in the configuration file. If this
option is absent, the server password is read from standard input.
–A admin_id Specifies the Tivoli Access Manager administrator name. If this option is not specified, sec_master is the default.

The ID is an alphanumeric, case-insensitive string. The minimum and maximum lengths of the ID, if there are limits, are imposed by the underlying registry. For Active Directory, the maximum length is 256 alphanumeric characters.

–t ssl_timeout Specifies the SSL session timeout in seconds. The ssl_timeout value must be in the range 1–86400. The default value is 7200.

–e pwd_life Sets the keyring file password expiration time in days. The pwd_life value is 1 to 7200 (days). To use the currently configured value, specify 0.

–l listening_mode Sets the listening-enabled flag in the configuration file. The value of this option must be yes or no. If not specified, the default is no. A value of yes requires that the –r option have non-zero value.

–a refresh_mode Sets the certificate and keyring file password auto-refresh enabled flag in the configuration file. The default value is yes.

–C cert_file Specifies the fully qualified name of the file containing the base-64 encoded SSL certificate used when the server authenticates directly with the user registry.

–h host_name Specifies the TCP host name used by the policy server to contact this server. This name is saved in the configuration file using the azn-app-host key. The default is the local host name returned by the operating system.

Valid values for host_name include any valid IP host name.

Examples:
host = libra
host = libra.dallas.ibm.com

Availability
This command is located in the following default installation directories:

- On UNIX systems:
  /opt/PolicyDirector/bin/
- On Windows systems:
  c:\Program Files\Tivoli\Policy Director\bin\

Return Codes
The following exit status codes can be returned:

0 The command completed successfully.
1 The command failed.
svrsslcfg –modify
Changes an application server’s current configuration.

Syntax
svrsslcfg –modify –f cfg_file [–t ssl_timeout] [–C cert_file] [–l listening_mode]

Options
–f cfg_file
Specifies the configuration file path and name. A valid file name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Note: For Windows, file names cannot have these characters: a forward slash (/), a backward slash (\), a colon (:), a question mark (?), double quotation marks, and an asterisk (*).

For example:
UNIX
/opt/PolicyDirector/etc/activedir.conf
Windows
C:\Program Files\Tivoli\Policy Director\etc\activedir.conf

–t ssl_timeout
Specifies the SSL session timeout in seconds. The ssl_timeout value must be in the range 1–86400.

–C cert_file
Specify the fully qualified name of the file containing the base-64 encoded SSL certificate used when the server authenticates directly with the user registry.

–l listening_mode
Sets the listening-enabled flag in the configuration file. Values are yes and no. A value of yes requires that the listening port number in the configuration file be non-zero.

Comments
Stop the Tivoli Access Manager policy server before running this command.

Availability
This command is located in the following default installation directories:
• On UNIX systems: /opt/PolicyDirector/bin/
• On Windows systems: c:\Program Files\Tivoli\Policy Director\bin

Return Codes
The following exit status codes can be returned:
0 The command completed successfully.
1 The command failed.
svrsslcfg –rmv_replica

Removes an authorization server replica from an application server’s configuration.

Syntax

svrsslcfg –rmv_replica –f cfg_file –h host_name

Options

–f cfg_file

Specifies the configuration file path and name.
A valid file name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-Z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

Note: For Windows, file names cannot have these characters: a forward slash (/), a backward slash (\), a colon(:), a question mark (?), double quotation marks, and an asterisk (*).

For example:

UNIX /opt/PolicyDirector/etc/activedir.conf

Windows C:\Program Files\Tivoli\PolicyDirector\etc\activedir.conf

–h host_name

Specifies the TCP host name of an authorization server replica.
Valid values for host_name include any valid IP host name.

Examples:

host = libra
host = libra.dallas.ibm.com

Availability

This command is located in the following default installation directories:

• On UNIX systems:
  /opt/PolicyDirector/bin/
• On Windows systems:
  c:\Program Files\Tivoli\Policy Director\bin\

Return Codes

The following exit status codes can be returned:

0 The command completed successfully.
1 The command failed.
svrsslcfg –unconfig

Unconfigures an application server. The key ring files are deleted and the server is removed from the user registry and Tivoli Access Manager database.

Syntax

svrsslcfg –unconfig –f \[cfg_file\] –n appl_name \[-P admin_pwd\] \[-A admin_id\]

Options

- `–f cfg_file` Specifies the configuration path and file name.
  A valid file name is an alphanumeric string that can be specified using any case. Valid string characters are the letters a-z, the numbers 0-9, a period (.), a comma (,), an equal sign (=), a forward slash (/), an underscore (_), a plus sign (+), a hyphen (-), an at sign (@), an ampersand (&), and an asterisk (*).

  **Note:** For Windows, file names cannot have these characters: a forward slash (/), a backward slash (/), a colon(:), a question mark (?), double quotation marks, and an asterisk (*).

  For example:

  UNIX /opt/PolicyDirector/etc/activedir.conf
  Windows C:\Program Files\Tivoli\Policy
                     Director\etc\activedir.conf

- `–n appl_name` Specifies the name of the application. The name is combined with the host name to create unique names for Tivoli Access Manager objects created for your application. The following names are reserved for Tivoli Access Manager applications: ivacld, secmgrd, ivnet, and ivweb.

- `–P admin_pwd` Specifies the Tivoli Access Manager administrator password. If this option is not specified, the password is read from standard in (stdin).

- `–A admin_id` Specifies the Tivoli Access Manager administrator name. The default is sec_master.

  The ID is an alphanumeric, case-insensitive string. The minimum and maximum lengths of the ID, if there are limits, are imposed by the underlying registry. For Active Directory, the maximum length is 256 alphanumeric characters.

Authorization

This command fails only if you are not authorized to run the command or the policy server could not be contacted. This command is designed to clean up partial or damaged configurations and so that errors for missing or invalid information are not reported.

Comments

Stop the server application before running this command.

Availability

This command is located in the following default installation directories:

- On UNIX systems:
  /opt/PolicyDirector/bin/
• On Windows systems:
  c:\Program Files\Tivoli\Policy Director\bin\

Return Codes
The following exit status codes can be returned:

0  The command completed successfully.

1  The command failed.
Appendix. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX
DB2
IBM
IBM logo
OS/390
SecureWay
Tivoli
Lotus and Domino are trademarks of International Business Machines Corporation and Lotus Development Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.
User registry differences

The following user registry differences are known to exist in this version of IBM Tivoli Access Manager (Tivoli Access Manager.)

1. Leading and trailing blanks in user names and group names are ignored when using LDAP or Microsoft Active Directory as the user registry in an Tivoli Access Manager secure domain. However, when using a Lotus Domino server as a user registry, leading and trailing blanks are significant. To ensure that processing is consistent regardless of what user registry is being used, define users and groups in the user registry without leading or trailing blanks in their names.

2. The forward slash character (/) should be avoided in user and group names defined using distinguished name strings. The forward slash character is treated differently in different user registries:

**Lotus Domino server**

Users and groups can not be created with names using a distinguished name string containing a forward slash character. To avoid the problem, either do not use a forward slash character or define the user without using the distinguished name designation:

```bash
pdadmin user create myuser username/locinfo test test testpwd
```

instead of using this one:

```bash
pdadmin user create myuser cn=username/o=locinfo test test testpwd
```

**Microsoft Active Directory**

Users and groups can be created with names using a distinguished name string containing a forward slash character. However, subsequent operations on the object might fail as some Active Directory functions interpret the forward slash character as a separator between the object name and the host name. To avoid the problem, do not use a forward slash character to define the user.

3. When using a multi-domain Microsoft Active Directory user registry, multiple users and groups can be defined with the same short name as long as they reside in different domains. To query information associated with a specific user or group, use the full name, including the domain, of the user or group to ensure that you are getting the correct information. If the domain information is omitted, information about the user or group defined in the default domain is returned, which might not be the expected user or group. The sole use of a short name to identify a user or group should be avoided for the same reason.

4. If Microsoft Active Directory is used as the user registry, care must be taken with user and group names that contain period characters (.) Active Directory does not permit a name to end with a period. (See Microsoft Knowledge Base article 316595 for details.) The first twenty (20) characters of a user or group name created by Tivoli Access Manager are mapped to a SAMAccountName in Active Directory. If the 20th character happens to be a period character, Active Directory considers the name not valid and generates an error. This can happen if a server in the Tivoli Access Manager happens to have a period in its name in that position, such as centralpolicyserver.company.com.

To avoid this problem, rename servers in the Tivoli Access Manager environment that have a period character in the 20th position of their name. Alternately, if the period occurs in the DNS suffix for a Microsoft Windows server, you might be able to avoid the problem by removing the primary DNS suffix from the Network settings.

5. When using iPlanet Version 5.0 as the user registry, a user that is created, added to a group, and then deleted from the user registry retains its group membership. If a user with the same name is created at some later time, the new user automatically inherits the old group membership and might be given inappropriate permissions. It is strongly recommended that the user be removed from all groups before the user is deleted. This problem does not occur when using the other supported user registries.

6. Attempting to add a duplicate user to a group produces different results based on the user registry being used. Table 13 on page 164 outlines the differences.
Table 13. User registry differences when adding a duplicate user to a group

<table>
<thead>
<tr>
<th>Operation</th>
<th>LDAP</th>
<th>Lotus Domino server</th>
<th>Microsoft Active Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add one user and that user is duplicate</td>
<td>Error</td>
<td>No error</td>
<td>Error</td>
</tr>
<tr>
<td>Add multiple users, first user is duplicate</td>
<td>Error for all users</td>
<td>No error</td>
<td>Error for all users</td>
</tr>
<tr>
<td>Add multiple users, a user other than the first is a duplicate</td>
<td>Error for all users</td>
<td>No error</td>
<td>Partial completion message</td>
</tr>
</tbody>
</table>

7. Attempting to remove a user from a group who is not a member of the group produces different results based on the user registry being used. [Table 14] outlines the differences.

Table 14. User registry differences when removing a user from a group who is not a member of the group

<table>
<thead>
<tr>
<th>Operation</th>
<th>LDAP</th>
<th>Lotus Domino server</th>
<th>Microsoft Active Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove one user, user is not in the group</td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
</tr>
<tr>
<td>Remove multiple users, first user not in the group</td>
<td>Error for all users</td>
<td>Error</td>
<td>Error for all users</td>
</tr>
<tr>
<td>Remove multiple users, a user other than the first is not in the group</td>
<td>Error for all users</td>
<td>Partial completion message</td>
<td>Partial completion message</td>
</tr>
</tbody>
</table>

8. The maximum lengths of various names associated with Tivoli Access Manager vary depending on the user registry being used. See [Table 15] for a comparison of the maximum lengths allowed and the recommended maximum length to use to ensure compatibility with all the user registries supported by Tivoli Access Manager.

Table 15. Maximum lengths for names based on user registry

<table>
<thead>
<tr>
<th>Maximum length of:</th>
<th>LDAP</th>
<th>Microsoft Active Directory</th>
<th>Lotus Domino server</th>
<th>Recommended maximum value</th>
</tr>
</thead>
<tbody>
<tr>
<td>First name (LDAP CN)</td>
<td>256</td>
<td>64</td>
<td>960</td>
<td>64</td>
</tr>
<tr>
<td>Middle name</td>
<td>128</td>
<td>64</td>
<td>65535</td>
<td>64</td>
</tr>
<tr>
<td>Last name (surname)</td>
<td>128</td>
<td>64</td>
<td>960</td>
<td>64</td>
</tr>
<tr>
<td>Registry UID (LDAP DN)</td>
<td>1024</td>
<td>2048</td>
<td>255</td>
<td>This value is user registry-specific and must be changed when changing user registries.</td>
</tr>
<tr>
<td>Tivoli Access Manager user identity</td>
<td>256</td>
<td>2048 - 1 - length_of_domain_name</td>
<td>200 - 4 - length_of_domain_name</td>
<td>This value is user registry-specific and must be changed when changing user registries.</td>
</tr>
<tr>
<td>User password</td>
<td>unlimited</td>
<td>256</td>
<td>unlimited</td>
<td>256</td>
</tr>
<tr>
<td>User description</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>Group name</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>Group description</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>Single signon resource name</td>
<td>240</td>
<td>256</td>
<td>256</td>
<td>240</td>
</tr>
</tbody>
</table>
Table 15. Maximum lengths for names based on user registry (continued)

<table>
<thead>
<tr>
<th>Maximum length of:</th>
<th>LDAP</th>
<th>Microsoft Active Directory</th>
<th>Lotus Domino server</th>
<th>Recommended maximum value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single signon resource description</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>Single signon user ID</td>
<td>240</td>
<td>256</td>
<td>256</td>
<td>240</td>
</tr>
<tr>
<td>Single signon password</td>
<td>unlimited</td>
<td>256</td>
<td>unlimited</td>
<td>256</td>
</tr>
<tr>
<td>Single signon group name</td>
<td>240</td>
<td>256</td>
<td>256</td>
<td>240</td>
</tr>
<tr>
<td>Single signon group description</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>Action name</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Action description, action type</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
</tr>
<tr>
<td>Object name, object space name, ACL name, POP name</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
</tr>
<tr>
<td>Object description, object space description, ACL description, POP description</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
</tr>
</tbody>
</table>

Even though some names can be of unlimited length, excessive lengths can result in policy that is difficult to manage and might result in poor system performance. Choose maximum values that are logical for your environment.

9. Users created in a Lotus Domino server or Microsoft Active Directory user registry are automatically given the capability to own single signon credentials and this capability can not be removed. When using an LDAP user registry, this capability must be explicitly granted to a user and subsequently can be removed.

10. When the Tivoli Access Manager policy server is using either Microsoft Active Directory or a Lotus Domino server as its user registry, existing Tivoli SecureWay Policy Director, Version 3.8 clients are not able to connect to the policy server. Either use a different user registry or upgrade the clients to Tivoli Access Manager.
access control. In computer security, the process of ensuring that the resources of a computer system can be accessed only by authorized users in authorized ways.

access control list (ACL). In computer security, a list that is associated with an object that identifies all the subjects that can access the object and their access rights. For example, an access control list is a list that is associated with a file that identifies the users who can access the file and identifies the users’ access rights to that file.

access permission. The access privilege that applies to the entire object.

action. An access control list (ACL) permission attribute. See also access control list.

ACL. See access control list.

administration service. An authorization API runtime plug-in that can be used to perform administration requests on a Tivoli Access Manager resource manager application. The administration service will respond to remote requests from the pdadmin command to perform tasks, such as listing the objects under a particular node in the protected object tree. Customers may develop these services using the authorization ADK.

attribute list. A linked list that contains extended information that is used to make authorization decisions. Attribute lists consist of a set of name = value pairs.

authentication. (1) In computer security, verification of the identity of a user or the user’s eligibility to access an object. (2) In computer security, verification that a message has not been altered or corrupted. (3) In computer security, a process that is used to verify the user of an information system or of protected resources. See also multi-factor authentication, network-based authentication, and step-up authentication.

authorization. (1) In computer security, the right granted to a user to communicate with or make use of a computer system. (2) The process of granting a user either complete or restricted access to an object, resource, or function.

authorization rule. See rule.

authorization service plug-in. A dynamically loadable library (DLL or shared library) that can be loaded by the Tivoli Access Manager authorization API runtime client at initialization time in order to perform operations that extend a service interface within the Authorization API.

The service interfaces that are currently available include Administration, External Authorization, Credentials modification, Entitlements and PAC manipulation interfaces. Customers may develop these services using the authorization ADK.
HTTP request, and vice versa. A CGI script is a CGI program that is written in a scripting language, such as Perl.

configuration. (1) The manner in which the hardware and software of an information processing system are organized and interconnected. (2) The machines, devices, and programs that make up a system, subsystem, or network.

collection. (1) In data communication, an association established between functional units for conveying information. (2) In TCP/IP, the path between two protocol applications that provides reliable data stream delivery service. In the Internet, a connection extends from a TCP application on one system to a TCP application on another system. (3) In system communications, a line over which data can be passed between two systems or between a system and a device.

container object. A structural designation that organizes the object space into distinct functional regions.

cookie. Information that a server stores on a client machine and accesses during subsequent sessions. Cookies allow servers to remember specific information about clients.

credentials. Detailed information, acquired during authentication, that describes the user, any group associations, and other security-related identity attributes. Credentials can be used to perform a multitude of services, such as authorization, auditing, and delegation.

credentials modification service. An authorization API runtime plug-in which can be used to modify a Tivoli Access Manager credential. Credentials modification services developed externally by customers are limited to performing operation to add and remove from the credentials attribute list and only to those attributes that are considered modifiable.

cross domain authentication service (CDAS). A WebSEAL service that provides a shared library mechanism that allows you to substitute the default WebSEAL authentication mechanisms with a custom process that returns a Tivoli Access Manager identity to WebSEAL. See also WebSEAL.

cross domain mapping framework (CDMF). A programming interface that allows a developer to customize the mapping of user identities and the handling of user attributes when WebSEAL e-Community SSO function are used.

D

daemon. A program that runs unattended to perform continuous or periodic systemwide functions, such as network control. Some daemons are triggered automatically to perform their task; others operate periodically.

directory schema. The valid attribute types and object classes that can appear in a directory. The attribute types and object classes define the syntax of the attribute values, which attributes must be present, and which attributes may be present for the directory.

distinguished name (DN). The name that uniquely identifies an entry in a directory. A distinguished name is made up of attribute:value pairs, separated by commas.

digital signature. In e-commerce, data that is appended to, or is a cryptographic transformation of, a data unit that enables the recipient of the data unit to verify the source and integrity of the unit and to recognize potential forgery.

domain. (1) A logical grouping of users, systems, and resources that share common services and usually function with a common purpose. (2) That part of a computer network in which the data processing resources are under common control. See also domain name.

domain name. In the Internet suite of protocols, a name of a host system. A domain name consists of a sequence of subnames that are separated by a delimiter character. For example, if the fully qualified domain name (FQDN) of a host system is as400.rchland.vnet.ibm.com, each of the following is a domain name: as400.rchland.vnet.ibm.com, vnet.ibm.com, ibm.com.

E

EAS. See External Authorization Service.

cryptography. In computer security, the process of transforming data into an unintelligible form in such a way that the original data either cannot be obtained or can be obtained only by using a decryption process.

entitlement. A data structure that contains externalized security policy information. Entitlements contain policy data or capabilities that are formatted in a way that is understandable to a specific application.

entitlement service. An authorization API runtime plug-in which can be used to return entitlements from an external source for a principal or set of conditions. Entitlements are normally application specific data that will be consumed by the resource manager application in some way or added to the principal’s credentials for use further on in the authorization process. Customers may develop these services using the authorization ADK.
external authorization service. An authorization API runtime plug-in that can be used to make application or environment specific authorization decisions as part of the Tivoli Access Manager authorization decision chain. Customers may develop these services using the authorization ADK.

F

file transfer protocol (FTP). In the Internet suite of protocols, an application layer protocol that uses Transmission Control Protocol (TCP) and Telnet services to transfer bulk-data files between machines or hosts.

global signon (GSO). A flexible single sign-on solution that enables the user to provide alternative user names and passwords to the back-end Web application server. Global signon grants users access to the computing resources they are authorized to use — through a single login. Designed for large enterprises consisting of multiple systems and applications within heterogeneous, distributed computing environments, GSO eliminates the need for users to manage multiple user names and passwords. See also single signon.

G

host. A computer that is connected to a network (such as the Internet or an SNA network) and provides an access point to that network. Also, depending on the environment, the host may provide centralized control of the network. The host can be a client, a server, or both a client and a server simultaneously.

HTTP. See Hypertext Transfer Protocol.

hypertext transfer protocol (HTTP). In the Internet suite of protocols, the protocol that is used to transfer and display hypertext documents.

I

Internet protocol (IP). In the Internet suite of protocols, a connectionless protocol that routes data through a network or interconnected networks and acts as an intermediary between the higher protocol layers and the physical network.

Internet suite of protocols. A set of protocols developed for use on the Internet and published as Requests for Comments (RFCs) through the Internet Engineering Task Force (IETF).

interprocess communication (IPC). (1) The process by which programs communicate data to each other and synchronize their activities. Semaphores, signals, and internal message queues are common methods of interprocess communication. (2) A mechanism of an operating system that allows processes to communicate with each other within the same computer or over a network.

IP. See Internet Protocol.

IPC. See Interprocess Communication.

J

junction. An HTTP or HTTPS connection between a front-end WebSEAL server and a back-end Web application server. WebSEAL uses a junction to provide protective services on behalf of the back-end server.

K

key. In computer security, a sequence of symbols that is used with a cryptographic algorithm for encrypting or decrypting data. See private key and public key.

key database file. See key ring.

key file. See key ring.

key pair. In computer security, a public key and a private key. When the key pair is used for encryption, the sender uses the public key to encrypt the message, and the recipient uses the private key to decrypt the message. When the key pair is used for signing, the signer uses the private key to encrypt a representation of the message, and the recipient uses the public key to decrypt the representation of the message for signature verification.

key ring. In computer security, a file that contains public keys, private keys, trusted roots, and certificates.

L


lightweight directory access protocol (LDAP). An open protocol that (a) uses TCP/IP to provide access to directories that support an X.500 model and (b) does not incur the resource requirements of the more complex X.500 Directory Access Protocol (DAP). Applications that use LDAP (known as directory-enabled applications) can use the directory as a common data store and for retrieving information about people or services, such as e-mail addresses, public keys, or service-specific configuration parameters. LDAP was originally specified in RFC 1777. LDAP version 3 is specified in RFC 2251, and the IETF continues work on additional standard functions. Some of the IETF-defined standard schemas for LDAP are found in RFC 2256.
lightweight third party authentication (LTPA).  An authentication framework that allows single sign-on across a set of Web servers that fall within an Internet domain.

LTPA.  See lightweight third party authentication.

M

management domain.  The default domain in which Tivoli Access Manager enforces security policies for authentication, authorization, and access control. This domain is created when the policy server is configured. See also domain.

management server.  Obsolete. See policy server.

metadata.  Data that describes the characteristics of stored data.

migration.  The installation of a new version or release of a program to replace an earlier version or release.

multi-factor authentication.  A protected object policy (POP) that forces a user to authenticate using two or more levels of authentication. For example, the access control on a protected resource can require that the users authenticate with both user name/password and user name/token passcode. See also protected object.

multiplexing proxy agent (MPA).  A gateway that accommodates multiple client access. These gateways are sometimes known as Wireless Access Protocol (WAP) gateways when clients access a secure domain using a WAP. Gateways establish a single authenticated channel to the originating server and tunnel all client requests and responses through this channel.

N

network-based authentication.  A protected object policy (POP) that controls access to objects based on the internet protocol (IP) address of the user. See also protected object policy.

P

PAC.  See privilege attribute certificate.

permission.  The ability to access a protected object, such as a file or directory. The number and meaning of permissions for an object are defined by the access control list (ACL). See also access control list.

policy.  A set of rules that are applied to managed resources.

policy server.  The Tivoli Access Manager server that maintains the location information about other servers in the secure domain.

polling.  The process by which databases are interrogated at regular intervals to determine if data needs to be transmitted.

POP.  See protected object policy.

portal.  An integrated Web site that dynamically produces a customized list of Web resources, such as links, content, or services, available to a specific user, based on the access permissions for the particular user.

privilege attribute certificate.  A digital document that contains a principal's authentication and authorization attributes and a principal's capabilities.

privilege attribute certificate service.  An authorization API runtime client plug-in which translates a PAC of a predetermined format in to a Tivoli Access Manager credential, and vice-versa. These services could also be used to package or marshall a Tivoli Access Manager credential for transmission to other members of the secure domain. Customers may develop these services using the authorization ADK. See also privilege attribute certificate.

protected object.  The logical representation of an actual system resource that is used for applying ACLs and POPs and for authorizing user access. See also protected object policy and protected object space.

protected object policy (POP).  A type of security policy that imposes additional conditions on the operation permitted by the ACL policy to access a protected object. It is the responsibility of the resource manager to enforce the POP conditions. See also access control list, protected object, and protected object space.

protected object space.  The virtual object representation of actual system resources that is used for applying ACLs and POPs and for authorizing user access. See also protected object and protected object policy.

private key.  In computer security, a key that is known only to its owner. Contrast with public key.

public key.  In computer security, a key that is made available to everyone. Contrast with private key.

Q

quality of protection.  The level of data security, determined by a combination of authentication, integrity, and privacy conditions.

R

registry.  The datastore that contains access and configuration information for users, systems, and software.
replica. A server that contains a copy of the directory or directories of another server. Replicas back up servers in order to enhance performance or response times and to ensure data integrity.

resource object. The representation of an actual network resource, such as a service, file, and program.

response file. A file that contains a set of predefined answers to questions asked by a program and that is used instead of entering those values one at a time.

role activation. The process of applying the access permissions to a role.

role assignment. The process of assigning a role to a user, such that the user has the appropriate access permissions for the object defined for that role.

routing file. An ASCII file that contains commands that control the configuration of messages.

RSA encryption. A system for public-key cryptography used for encryption and authentication. It was invented in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman. The system’s security depends on the difficulty of factoring the product of two large prime numbers.

rule. One or more logical statements that enable the event server to recognize relationships among events (event correlation) and to execute automated responses accordingly.

run time. The time period during which a computer program is executing. A runtime environment is an execution environment.

S

scalability. The ability of a network system to respond to increasing numbers of users who access resources.

schema. The set of statements, expressed in a data definition language, that completely describe the structure of a database. In a relational database, the schema defines the tables, the fields in each table, and the relationships between fields and tables.

secure sockets layer (SSL). A security protocol that provides communication privacy. SSL enables client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery. SSL was developed by Netscape Communications Corp. and RSA Data Security, Inc.

security management. The management discipline that addresses an organization's ability to control access to applications and data that are critical to its success.

self-registration. The process by which a user can enter required data and become a registered Tivoli Access Manager user, without the involvement of an administrator.

service. Work performed by a server. A service can be a simple request for data to be sent or stored (as with file servers, HTTP servers, e-mail servers, and finger servers), or it can be more complex work such as that of print servers or process servers.

silent installation. An installation that does not send messages to the console but instead stores messages and errors in log files. Also, a silent installation can use response files for data input. See also response file.

single signon (SSO). The ability of a user to logon once and access multiple applications without having to logon to each application separately. See also global signon.

SSL. See Secure Sockets Layer.

SSO. See Single Signon.

step-up authentication. A protected object policy (POP) that relies on a preconfigured hierarchy of authentication levels and enforces a specific level of authentication according to the policy set on a resource. The step-up authentication POP does not force the user to authenticate using multiple levels of authentication to access any given resource but requires the user to authenticate at a level at least as high as that required by the policy protecting a resource.

suffix. A distinguished name that identifies the top entry in a locally held directory hierarchy. Because of the relative naming scheme used in Lightweight Directory Access Protocol (LDAP), this suffix applies to every other entry within that directory hierarchy. A directory server can have multiple suffixes, each identifying a locally held directory hierarchy.

T

token. (1) In a local area network, the symbol of authority passed successively from one data station to another to indicate the station temporarily in control of the transmission medium. Each data station has an opportunity to acquire and use the token to control the medium. A token is a particular message or bit pattern that signifies permission to transmit. (2) In local area networks (LANs), a sequence of bits passed from one device to another along the transmission medium. When the token has data appended to it, it becomes a frame.

trusted root. In the Secure Sockets Layer (SSL), the public key and associated distinguished name of a certificate authority (CA).
**U**

**uniform resource identifier (URI).** The character string used to identify content on the Internet, including the name of the resource (a directory and file name), the location of the resource (the computer where the directory and file name exist), and how the resource can be accessed (the protocol, such as HTTP). An example of a URI is a uniform resource locator, or URL.

**uniform resource locator (URL).** A sequence of characters that represent information resources on a computer or in a network such as the Internet. This sequence of characters includes (a) the abbreviated name of the protocol used to access the information resource and (b) the information used by the protocol to locate the information resource. For example, in the context of the Internet, these are abbreviated names of some protocols used to access various information resources: http, ftp, gopher, telnet, and news; and this is the URL for the IBM home page: http://www.ibm.com.

**URI.** See *uniform resource identifier.*

**URL.** See *uniform resource locator.*

**user.** Any person, organization, process, device, program, protocol, or system that uses a service provided by others.

**user registry.** See *registry.*

**V**

**virtual hosting.** The capability of a Web server that allows it to appear as more than one host to the Internet.

**W**

**Web Portal Manager (WPM).** A Web-based graphical application used to manage Tivoli Access Manager Base and WebSEAL security policy in a secure domain. An alternative to the *pdadmin* command line interface, this GUI enables remote administrator access and enables administrators to create delegated user domains and assign delegate administrators to these domains.

**WebSEAL.** A Tivoli Access Manager blade. WebSEAL is a high performance, multi-threaded Web server that applies a security policy to a protected object space. WebSEAL can provide single sign-on solutions and incorporate back-end Web application server resources into its security policy.

**WPM.** See *Web Portal Manager.*
Index

A
access control list (ACL) commands
  acl attach 5, 10
  acl create 5, 11
  acl delete 5, 12
  acl detach 5, 13
  acl find 5, 14
  acl list 5, 15
  acl modify 5, 16
  acl show 5, 21
action
  group 26
action commands
  action create 6, 23
  action delete 6, 25
  action group 6, 26
  action list 6, 27
attach
  access control list (ACL) 10
  protected object policy (POP) 66

B
basslcfg
  change password 126
  configure 127
  get certificate 128
  modify 129
  ping 130

C
command modes
  interactive 2
  multiple 2
  single 1
command option processing 3
create
  access control list (ACL) 11
  actions 23
  group 31
  object 45
  object space 57
  protected object policy (POP) 68
  rsrcco 78
  rsrccred 83
  rsrccgroup 92
  user 113

D
delete
  access control list (ACL) 12
  actions 25
  group 33
  object 47
  object space command 59
  (continued)
  protected object policy (POP) 69
  rsrcco 80
  rsrccred 85
  rsrccgroup 93
  user 115
detach
  access control list (ACL) 13
  protected object policy (POP) 70
disallowed characters, GSO commands 3
domain commands
  domain create 6
  domain delete 6
  domain list 6
  domain modify 6
  domain show 6

F
find
  access control list (ACL) 14
  protected object policy (POP) 71

G
group commands
  group create 6, 31
  group delete 6, 33
  group import 6, 34
  group list 6, 35
  group modify 6, 37
  group show 6, 39
GSO commands
  disallowed characters 3

I
import
  group 34
  user 116
interactive command mode 1, 2

L
list
  access control list (ACL) 15
  actions 27
  group 35
  object 48
  object space command 60
  protected object policy (POP) 72

M
mgrsslcfg
  change certificate 134
  change password 135
mgrsslcfg (continued)
  configure 136
  modify 137
modify
  access control list (ACL) 16
  group 37
  object 52
  protected object policy (POP) 73
  rsrccred 88
  rsrcgroup 95
  user 120
multiple command mode 1, 2

O
object
  listandshow 50
object commands
  object create 7, 45
  object delete 7, 47
  object list 7, 48
  object listandshow 7, 50
  object modify 7, 52
  object show 7, 55
object space commands
  objectspace create 7, 57
  objectspace delete 7, 59
  objectspace list 7
objectspace commands
  objectspace list 60

P
pdadmin
  command option processing 3
  help 41
  login 43
  modes 1
  utility 1
pdadmin utilities
  exit command line mode 30
  logout 44
  quit command line mode 30
  show error message 29
pdbackup
  backs up, restores, and extracts data 138
pdconfig 141
pdjrtecfg
  configures Java runtime environment 142
policy commands
  policy get 7, 61
  policy set 7, 63
protected object policy (POP) commands
  pop attach 7, 66
  pop create 7, 68
  pop delete 7, 69
  pop detach 7, 70
  pop find 7, 71
  pop list 7, 72
  pop modify 7, 73
  pop show 7, 76

R
related publications ix
resource commands
  rsrc create 8, 78
  rsrc delete 8, 80
  rsrc list 8, 81
  rsrc show 8, 82
  rsrccred create 8, 83
  rsrccred delete 8, 85
  rsrccred list user 8, 87
  rsrccred modify 8, 88
  rsrccred show 8, 90
  rsrcgroup create 8, 92
  rsrcgroup delete 8, 93
  rsrcgroup list 8, 94
  rsrcgroup modify 8, 95
  rsrcgroup show 8, 97
rsrccred
  list user 87
rsrcgroup
  list 94

S
server
  list 98
  listtasks 99
  replicate 101
server commands
  admin show conf 28
  server list 8, 98
  server listtasks 8, 99
  server replicate 8, 101
  server show 8, 102
  server task 8, 104
  server task (WebSEAL) 105
  server task add (WebSEAL) 107
  server task stats (WebSEAL) 109
  server task trace 111
show
  access control list (ACL) 21
  group 39
  object
    listandshow 55
  protected object policy (POP) 76
  rsrccred 82
  rsrcgroup 97
  server 102
  user 122
single command mode 1
special disallowed characters 3
svrsslcfg
  add replica 146
  change certificate 149
  change password 152
  change port 151
  change replica 147
svrsslcfg (continued)
    configure 153
    modify 155
    remove replica 156
    unconfigure 157

U
user
    list 118
user commands
    user create 9, 113
    user delete 9, 115
    user import 9
    user import 116
    user list 9, 118
    user modify 9, 120
    user show 9, 122
user registry
    differences 163
    maximum values 164, 165
utilities
    basssslcfg –chgpwd 126
    basssslcfg –config 127, 131
    basssslcfg –getcacert 128
    basssslcfg –modify 129
    basssslcfg –ping 130
    install_pd rte 133
    mgrsslcfg –chgcert 134
    mgrsslcfg –chgpwd 135
    mgrsslcfg –config 136
    mgrsslcfg –modify 137
    pd_start 144
    pdbackup 138
    pdjrtecfg 142
    pdversion 145
    svrsslcfg –add_replica 146
    svrsslcfg –chg_repl ica 147
    svrsslcfg –chgcert 149
    svrsslcfg –chgpport 151
    svrsslcfg –chgpwd 152
    svrsslcfg –config 153
    svrsslcfg –modify 155
    svrsslcfg –rmv_repl ica 156
    svrsslcfg –unconfig 157