PGP CORPORATION

PGP KEYSERVER ENTERPRISE EDITION

Administrator's Guide

Version 7.0

Version Information

PGP Keyserver 7.0 Administrator's Guide. Released November, 2002.

Trademark Information

PGP and Pretty Good Privacy are registered trademarks of PGP Corporation in the U.S. and other countries. All other registered and unregistered trademarks in this document are the sole property of their respective owners.

Licensing and Patent Information

The IDEA™ cryptographic cipher described in U.S. patent number 5,214,703 is licensed from Ascom Tech AG. The CAST encryption algorithm is licensed from Northern Telecom, Ltd. PGP Corporation may have patents and/or pending patent applications covering subject matter in this software or its documentation; the furnishing of this software or documentation does not give you any license to these patents.

Acknowledgments

The compression code in PGP is by Mark Adler and Jean-Loup Gailly, used with permission from the free Info-ZIP implementation. This software is based in part on the work of the Independent JPEG Group. Soft TEMPEST font courtesy of Ross Anderson and Marcus Kuhn. Biometric word list for fingerprint verification courtesy of Patrick Juola. The Apple Mail plugin is based in part on the work of Stéphane Corthésy and Sen:te, and is used with permission.

Export Information

Export of this software and documentation may be subject to compliance with the rules and regulations promulgated from time to time by the Bureau of Export Administration, United States Department of Commerce, which restrict the export and re-export of certain products and technical data.

Limitations

The software provided with this documentation is licensed to you for your individual use under the terms of the End User License Agreement provided with the software. The information in this document is subject to change without notice. PGP Corporation does not warrant that the information meets your requirements or that the information is free of errors. The information may include technical inaccuracies or typographical errors. Changes may be made to the information and incorporated in new editions of this document, if and when made available by PGP Corporation.

About PGP Corporation

PGP Corporation, the recognized worldwide leader in secure messaging and data storage, builds products that allow corporations to ensure confidential customer and individual information remains secure. Over the last 10 years, PGP technology has developed a global reputation for enabling open, trusted, and highly reliable security products. PGP has thousands of corporate/government users and millions of individual users worldwide, including many of the world's largest and most security sensitive enterprises, government agencies, individuals, and cipher experts. Contact PGP Corporation at www.pgp.com or toll free at 866.747.5483 (866.PGP.LIVE).

Table of Contents

Introduction

In today's world of electronic interconnection and information accessibility, a growing concern is information security. Maintaining the privacy, integrity, and authenticity of information that is electronically stored or transmitted is a fundamental challenge in the information technology community.

Your PGP Keyserver software takes you a step in the direction of improved data security. By installing PGP Keyserver software on a machine in your network, you can store and retrieve public data encryption keys generated by PGP.

While keys can be stored on public keyservers, having your own keyserver allows you to control and manage the keys stored on the keyserver. You can implement requirements—such as certification of all keys by a trusted employee—that increase your confidence in the legitimacy of keys stored on the keyserver.

This guide describes how to install, configure, operate, and maintain PGP Keyserver.

Who should read this guide

This guide is for system administrators or others who are responsible for setting up and running PGP Keyserver. PGP Keyserver allows PGP users to submit and retrieve keys according to the policies enforced at your site.

Licensing

PGP uses a license number system to determine what PGP features will be active on your computer. For complete information and purchase options, go to https://store.pgp.com.

Note: PGP Keyserver requires an Enterprise license.

This Administrator's Guide

The chapters and appendices in this Administrator's Guide include:

- Chapter 1, About PGP Keyserver, provides a brief description of public key cryptography, as well as an overview of how PGP Keyserver is used to store and retrieve cryptographic keys.
- Chapter 2, Installing PGP Keyserver, describes how to install PGP Keyserver.
- Chapter 3, Getting Started, describes how to access PGP Keyserver's Web Console and configure the Replication Engine.
- Chapter 4, Controlling the Keyserver's Components, describes how to start PGP Keyserver, the Web Console, and Replication Engine.
- Chapter 5, Using the PGP Keyserver and Replication Engine, describes how to use PGP Keyserver and the Replication Engine, including performing searches on PGP Keyserver, adding keys, resolving keys in the pending bucket, and reviewing the status of PGP Keyserver and the Replication Engine. This chapter also includes a description of each of the Web Console's configuration panels (Select Configuration File, General, Database, Secure Mode, Access, Policy, and Replication).
- Chapter 6, Keyserver and Engine Configuration Settings, describes the replication process, PGP Keyserver and Replication Engine configurations, and how to install a new PGP Keyserver.
- Chapter 7, The Replication Process, describes how to change the PGP Keyserver's and Replication Engine's configuration settings and describes each configuration setting (General, Database, Secure Mode, Access, Policy, and Replication).
- Appendix A, LDAP Error Messages, includes a complete list of LDAP error messages found in the Access Log File.
- Appendix B, Receiving Keys from MIT-Style Keyservers, describes how to configure PGP Keyserver so that it can receive keys contained in a sync mail message from MIT-style keyservers.
- Appendix C, HTTP Support for PGP 5.0 Clients, describes the HTTP-to-LDAP gateway that allows existing PGP 5.0 clients to access a PGP Keyserver, which is included with PGP Keyserver.
- Appendix D, Keyserver and Engine Command Line Switches, contains listings of all PGP Keyserver and Replication Engine command line switches.

There is also a Glossary and an Index.

Recommended readings

This section identifies Web sites, books, and periodicals about the history, technical aspects, and politics of cryptography, as well as trusted PGP download sites.

The history of cryptography

- *The Code Book: The Evolution of Secrecy from Mary, Queen of Scots, to Quantum Cryptography*, Simon Singh, Doubleday & Company, Inc., 1999, ISBN 0-385-49531-5.
- *The Codebreakers: The Story of Secret Writing*, David Kahn, Simon & Schuster Trade, 1996, ISBN 0-684-83130-9 (updated from the 1967 edition). This book is a history of codes and code breakers from the time of the Egyptians to the end of WWII. Kahn first wrote it in the sixties; this is the revised edition. This book won't teach you anything about how cryptography is done, but it has been the inspiration of the whole modern generation of cryptographers.
- Aegean Park Press, www.aegeanparkpress.com. The Aegean Park Press publishes a number of interesting historic books ranging from histories (such as "The American Black Chamber," an exposé of U.S. cryptography during and after WWI) to declassified government documents.

Technical aspects of cryptography

Web sites

- www.iacr.org. International Association for Cryptologic Research (IACR). The IACR holds cryptographic conferences and publishes journals.
- www.pgpi.org. An international PGP Web site, which is not maintained by PGP Corporation, is an unofficial yet comprehensive resource for PGP.
- www.nist.gov/aes. The National Institute of Standards and Technology (NIST) Advanced Encryption Standard (AES) Development Effort, perhaps the most interesting project going on in cryptography today.
- www.ietf.org/rfc/rfc2440.txt. The IETF OpenPGP specification, written by Jon Callas, Lutz Donnerhacke, Hal Finney, and Rodney Thayer.
- www.ietf.org/rfc/rfc3156.txt. The IETF OpenPGP/MIME specification, written by Michael Elkins, Dave del Torto, Raph Levien, and Thomas Roessler.

Books and periodicals

- *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd edition, Bruce Schneier, John Wiley & Sons, 1996; ISBN 0-471-12845-7. If you can only buy one book to get started in cryptography, this is the one to buy.
- *Handbook of Applied Cryptography*, Alfred Menezes, Paul van Oorschot and Scott Vanstone, CRC Press, 1996; ISBN 0-8493-8523-7. This is the technical book you should get after Schneier. There is a lot of heavy-duty math in this book, but it is nonetheless usable for those who do not understand the math.
- *Journal of Cryptology*, International Association for Cryptologic Research (IACR). See www.iacr.org.
- *Advances in Cryptology*, conference proceedings of the IACR CRYPTO conferences, published yearly by Springer-Verlag. See www.iacr.org.
- *The Twofish Encryption Algorithm: A 128-Bit Block Cipher*, Bruce Schneier, et al, John Wiley & Sons, Inc., 1999; ISBN: 0471353817. Contains details about the Twofish cipher ranging from design criteria to cryptanalysis of the algorithm.

Politics of cryptography

Web sites

- www.epic.org, Electronic Privacy Information Center.
- www.crypto.org, Internet Privacy Coalition.
- www.eff.org, Electronic Frontier Foundation.
- www.privacy.org, privacy.org. Great information resource about privacy issues.
- www.cdt.org, Center for Democracy and Technology.
- www.philzimmermann.com, Phil Zimmermann's home page, his Senate testimony, and so on.

Books

- *Privacy on the Line: The Politics of Wiretapping and Encryption*, Whitfield Diffie and Susan Landau, The MIT Press, 1998, ISBN 0-262-04167-7. This book is a discussion of the history and policy surrounding cryptography and communications security. It is an excellent read, even for beginners and non-technical people. Includes information that even a lot of experts don't know.
- *Crypto: How the Code Rebels Beat the Government--Saving Privacy in the Digital Age*, Steven Levy, Penguin USA, 2001; ISBN 0140244328.

Network security

Books

- *Building Internet Firewalls*, Elizabeth D. Zwicky, D. Brent Chapman, Simon Cooper, and Deborah Russell (Editor), O'Reilly & Associates, Inc., 2000; ISBN: 1565928717. This book is a practical guide to designing, building, and maintaining firewalls.
- *Firewalls and Internet Security: Repelling the Wily Hacker*, William R. Cheswick, Steven M. Bellovin, Addison Wesley Longman, Inc., 1994; ISBN: 0201633574. This book is a practical guide to protecting networks from hacker attacks through the Internet. Available on the Web at www.wilyhacker.com.
- *Network Security: Private Communication in a Public World*, Second Edition, Charles Kaufman, Radia Perlman, and Mike Speciner, Pearson Education, 2002; ISBN: 0130460192. This book describes many network protocols, including Kerberos, IPsec, SSL, and others. It includes some basics of cryptography and works up from there to show how actual systems are constructed.

Symbols

Notes, Cautions, and Warnings are used in the following ways.

Notes are extra, but important, information.

Note: A Note adds important information, but you could still use the product if you didn't have that information.

Cautions indicate the possibility of loss of data or minor damage to equipment.

Caution: A Caution tells you about a situation with the potential for loss of data or minor damage to equipment. Special attention should be paid to Cautions.

Warnings indicate the possibility of significant damage to equipment or injury to human beings.

Warning: A Warning means that your equipment may be damaged or someone could be injured. Please take Warnings seriously.

1 About PGP Keyserver

Securing information that is stored or transmitted electronically is a fundamental challenge of the information technology community today. The principal approach to addressing this information security challenge is cryptography—the science of using mathematics to encrypt and decrypt data.

This chapter begins with a brief description of how cryptography works, how cryptographic keys are used, and how keyservers in general are used for storage and retrieval of cryptographic keys. The chapter then explains the features of PGP Keyserver software in particular, and how it works.

Note: For a more in-depth discussion of cryptography, refer to *An Introduction to Cryptography,* which is included with your product.

What is a cryptographic key?

Data encryption converts data that can be read (plaintext) into scrambled text that cannot be read (ciphertext). A *cryptographic key* is a number that is used in the process of encrypting or decrypting data. Larger numbers are harder to figure out—so larger cryptographic keys provide stronger data security.

The scrambling process of data encryption involves taking the plaintext and feeding it, along with the cryptographic key, to a mathematical function called a cryptographic algorithm. The ciphertext that the cryptographic algorithm spits out depends on both pieces of input: the plaintext and the key. Without the key, one cannot retrieve the original plaintext.

As the importance of preserving data privacy has increased, so has the complexity of the data encryption process. Conventional cryptography uses the same key for both encryption and decryption—it is referred to as a *symmetric* scheme. A more secure encryption method is *public key cryptography,* an *asymmetric* scheme that uses a pair of keys. A *public key* is used for encrypting data, and a corresponding *private,* or *secret, key* is used for decrypting data.

Those who want to communicate securely and privately keep a private key to themselves, but they publish their public key to the world. Anyone with a copy of the public key can then encrypt information that only the intended recipient with the corresponding private key can read. This scheme minimizes the likelihood of a key falling into the wrong hands.

The process of creating a private key must ensure that the key is unique and, as the name says, private. For example, users may be asked to move their mouse around to generate random motions that cannot be reproduced. These motions are then translated into a unique cryptographic key. A user's private key is typically stored encrypted with a passphrase on the user's computer.

What is a keyserver?

A *keyserver* is a computer that processes requests for storing and retrieving public keys. As the activity of encrypting data increases, so does the number of public keys. keyservers help to organize the processes of storing and then finding keys. In addition, a keyserver usually provides some administrative features that enable a company to maintain its security policies—for example, allowing only those keys that meet certain requirements to be stored.

One way to distribute your public key to others is to send it to a keyserver. Other ways to distribute your public key include storing the key on a diskette and giving it to specific people, or sending the key via email.

Giving someone your public key on a diskette is a way of assuring the person that the public key truly belongs to you. Verifying the authenticity of public keys is important; this is because a corrupt person who wants to read your mail can create a phony public-private keypair and post the public key, leading the world to believe that it is yours.

So how can public keys stored on keyservers be protected from this risk? Enter *digital certificates*. A digital certificate is a public key plus information that helps others verify that the key is genuine or valid. Digital certificates (also called *public key signatures*, or just *certificates*, for short) are used to thwart attempts to substitute one person's key for another.

A digital certificate consists of three things:

- A public key
- Certificate information ("identity" information about the user, such as name and email address)
- One or more digital signatures

A *digital signature* certifies that the signer vouches for the owner of the key. This is typically done after personally confirming that the key's user information matches the user's identity—this confirmation might be done by passport or personal knowledge. For more information about digital signatures, refer to *An Introduction to Cryptography*, included as part of your product package.

While the primary advantage of storing and distributing public keys using keyservers is convenience, the legitimacy of keys stored on keyservers is only as good as the signatures that certify them. A critical part of securing network communications effectively is ensuring the reliability of signatures that certify public keys.

What is PGP Keyserver?

PGP Keyserver is software installed on one or more machines. Throughout this documentation we shall refer to the machines themselves as keyservers.

keyservers are used primarily for storage and retrieval of public keys. In a typical corporate PGP implementation, employees store their public keys on the corporate keyserver. When any PGP user wants to exchange information with others by email, PGP retrieves the recipient's public key from the keyserver. Also, users can search the keyserver for particular keys that they can download and add to their personal *keyrings* (keyrings are files used for storing public keys).

PGP Keyserver data can be replicated to other keyservers. In this case, one machine is identified as the primary, or *master* keyserver, and the additional machines that have PGP Keyserver installed are called *slave* keyservers. The benefits to this arrangement are threefold:

- Increased fault tolerance: if one keyserver goes down, information can still be accessed from another keyserver.
- Greater load balancing: if one keyserver becomes overloaded with requests for information, some requests can be handled by another keyserver.
- Maximized access bandwidth: if a corporation has geographically distinct networks, each network can have its own keyserver. With this configuration, users avoid potentially low-speed interconnects between networks.

PGP Keyserver can also be used to store *key reconstruction data*—information used to re-create lost keys. When users generate a new key, they can create a set of simple questions and answers that are easy for them to remember and very difficult for anyone else to guess. The pre-defined questions are good examples of questions that work—they are very general, but they produce very personal answers. Users can also can also customize questions. If a key is lost or passphrase is forgotten, the information can be re-created if the user provides correct answers to the set of questions.

PGP Keyserver terminology

Web Console. The Web-based portion of the PGP Keyserver used to configure and monitor the PGP Keyserver.

Replication Engine. Keyserver software that replicates database entries to other Keyservers. The databases on these Keyservers are automatically updated to reflect the contents of the database on the primary PGP Keyserver.

Web server. Server software that serves up HTML documents, files, and scripts when requested by a client, such as a Web-browser. In this case, the software serves up the PGP Keyserver's Web Console and serves HTTP Keyserver requests.

Web Console port number. Port on the PGP Keyserver machine that clients use to connect to the Web Console via their Web browser.

Web Console configuration wizard. A script used to configure the PGP Keyserver's Web Console.

PGP Apache. The version of Apache shipped with PGP Keyserver which includes the ability to serve SSL connections.

keyserver. The machine where the PGP Keyserver resides.

HTTP Keyserver. An HTTP Interface included with the PGP Keyserver for backwards compatibility for PGP 5.0 clients.

PGP Keyserver's general features

PGP Keyserver's general features include the following:

- Automated installation and configuration of PGP Keyserver through easy-to-use scripts and a configuration wizard.
- Single point-of-control, web-based user interface—the Web Console—used for monitoring PGP Keyserver activities and making any required changes to its configuration.
- Flexible key retrieval that supports searches on multiple key attributes, such as the key type, creation date, and so on.
- Authentication safeguards that limit access to restricted PGP Keyserver functions, including access controls and signature verification.
- Monitoring features and log files that keep track of PGP Keyserver usage and provide data for analysis and planning purposes.
- PGP Replication Engine that allows you to maintain multiple PGP Keyservers with duplicate key databases.

Note: Throughout this documentation we may refer to the Replication Engine as the Engine.

How does PGP Keyserver work?

PGP Keyserver is designed to run on Windows and Sun Solaris platforms, and is based on the *Lightweight Directory Access Protocol (LDAP)*, a global directory model. LDAP provides a standard method to manage the submittal and retrieval of keys stored in a centralized database.

Installation and configuration

PGP Keyserver, including its Replication Engine and Web Console components, is installed using a simple-to-use installation wizard (Windows) or the package utility (Solaris). These tools make sure all of the software components are loaded in the proper sequence and stored in the appropriate directories.

If you install PGP Keyserver on multiple machines, you will use the **pgpexport** and **pgpexport** utilities, available from the Start menu, to copy the master Keyserver's database to the Keyservers running on the slave keyservers.

Thereafter, the Replication Engine sends key updates to other PGP Keyservers to keep the databases in sync.

Submitting keys

When you install PGP Keyserver, you define the set of rules, or the *certificate policy,* that Keyserver will use to enforce the acceptance, rejection, and retrieval of keys. When users attempt to place keys on or retrieve keys from the keyserver, PGP Keyserver enforces that policy.

When a key is submitted to the keyserver, PGP Keyserver checks to see if the key passes the policy requirements established during configuration.

For example, you might establish a policy that requires the PGP Keyserver to perform the following tasks:

- PGP Keyserver verifies that the key is signed by the required entities identified during Keyserver configuration.
- PGP Keyserver verifies the Authorized signatures on the key identified during Keyserver configuration.
- PGP Keyserver removes all unauthorized signatures or User IDs from the key before storing the key in the Keyserver database.

(For more information about certificate policy requirements, see "General configuration settings" on page 79.)

After enforcing the policy requirements, PGP Keyserver accepts the key. If the key does not pass the policy requirements, the key is rejected and a copy of the key is placed in a pending bucket. You can examine the key and decide if the key should be allowed on the keyserver. For more information about the pending bucket, see "Resolving keys in the pending bucket" on page 63.

Retrieving keys

When a key is placed on the keyserver, PGP users can retrieve the key to encrypt data and verify digital signatures.

All users can use the standard LDAP search and retrieval functions to access keys. Here are some of the attributes you can use in your search:

- email address
- user name (both first and last names)
- key IDs
- PGP key type, size, revocation status (that is, if the key's owner has revoked the key because it is old or compromised).
- creation and expiration dates

All users use the same LDAP interface to access keys (usually the PGP software). As System Administrator, your authority level, established during PGP Keyserver configuration, allows you to add, disable, and delete keys from the keyserver.

Importing and exporting keys

As System Administrator, you can import and export keys. Use these features to distribute large numbers of keys. You can import both PGP *keyrings* and *ASCII-armored key files* from any client machine that has proper access to the keyserver using the LDAP protocol. A *keyring* is a set of keys. An *ASCII-armored key file* is binary information encoded using a standard printable, 6-bit ASCII character set.

Key reconstruction

PGP Keyserver supports the Key Reconstruction feature available in PGP 7.0 and later.

Key reconstruction is used to re-create lost keys. When users generate a new key, they can create a set of simple questions and answers that are easy for them to remember and very difficult for anyone else to guess. The pre-defined questions are good examples of questions that work—they are very general, but they produce very personal answers. Users can also can also customize questions. If a key is lost or passphrase is forgotten, the information can be re-created if the user provides correct answers to the set of questions.

Replication of the database to other keyservers

The Replication Engine is a robust replication mechanism used to propagate the updates to a master PGP Keyserver's database to one or more slave keyservers. A replication daemon monitors the master keyserver and updates the slave keyservers' databases whenever a change occurs on the master keyserver.

You identify the slave keyservers when you run the Web Console. The Web Console stores this information in the master keyserver's configuration file.

Not all installations will use the master-slave keyserver configuration. A variety of PGP Keyserver configuration models are described in Chapter 6, Keyserver and Engine Configuration Settings.

PGPadmin preferences

PGP Keyserver now acts as a central repository for the PGP client software, storing PGPadmin settings. When the PGP administrator wants to send new settings to PGP users, the settings file can be placed on the PGP Keyserver for distribution.

For more information about PGPadmin settings and their distribution, please refer to the *PGP Administrator's Guide*.

2 Installing PGP Keyserver

This chapter describes how to install the PGP Keyserver, Enterprise Edition for Windows NT/2000 and UNIX software. Before you begin installing the PGP Keyserver, however, be sure to review the system requirements outlined below.

Note: Don't install the PGP client software on the machine onto which you are going to install your PGP Keyserver.

System Requirements

The PGP Keyserver consists of three components: a Keyserver (allows users to submit and retrieve keys from a database), a Replication Engine (an optional component that replicates changes to the Keyserver's database to other Keyservers), and a Web Console (allows you to control and monitor the Keyserver and Replication Engine via a Web browser).

To run the Web Console on Windows NT/2000 or UNIX you must have one of the following:

- Microsoft Internet Explorer 4.01 SP2 or later
- Netscape 4.x

To install the PGP Keyserver on a Windows NT/2000 Server, you must have:

- Windows NT version 4.0 Service Pack 6a or Windows 2000 Service Pack 3
- 64 MB RAM minimum
- 15 MB disk space for software
- 10 MB to 500 MB disk space for the certificate database (software requires between 6 and 10 kilobytes of disk space for each key stored on the Keyserver)
- Network interface card
- Any version of the PGP client software can be used for key management on a different machine.

To install the PGP Keyserver on a UNIX Server, you must have:

- Sun Solaris for SPARC (UNIX) version 2.6 or later
- 64 MB RAM minimum
- 30 MB disk space for software

- 10 MB to 500 MB disk space for the certificate database (software requires between 6 and 10 kilobytes of disk space for each key stored on the Keyserver)
- Network interface card
- PGP 7.0 is required for key management on the same machine. Any version of PGP can be used for key management on a different machine.

Note: The latest recommended patches from Sun are REQUIRED for Solaris 7 support. They can be obtained as a single patch bundle at the following Web site: http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access.

Installing on a Windows NT/2000 server

You can download the PGP Keyserver software from an authorized Web site or load the software from a CD-ROM. The self-extracting file, SETUP.EXE, automatically extracts and installs all of the necessary software components in their proper directory locations. After you install the software, run the Configuration wizard to customize the PGP Keyserver to meet the needs of your site.

Note: Don't install the PGP client software on the machine onto which you are going to install your PGP Keyserver.

To install the PGP Keyserver on a Windows NT/2000 server:

1. Start the Windows server.
2. Insert the PGP CD-ROM into the CD-ROM drive.
3. Double-click SETUP.EXE to start the Setup program.

Note: If you are installing from the CD-ROM, the Setup program automatically starts. If, however, the Setup program does not initiate, double-click SETUP.EXE in the Disk 1 folder on the CD-ROM.

The PGP Keyserver Welcome screen appears.

4. Review the information in the **Welcome** screen, then click **Next**.

   The license agreement appears.

5. Review the license agreement information, then click **Yes** to accept the licensing terms.

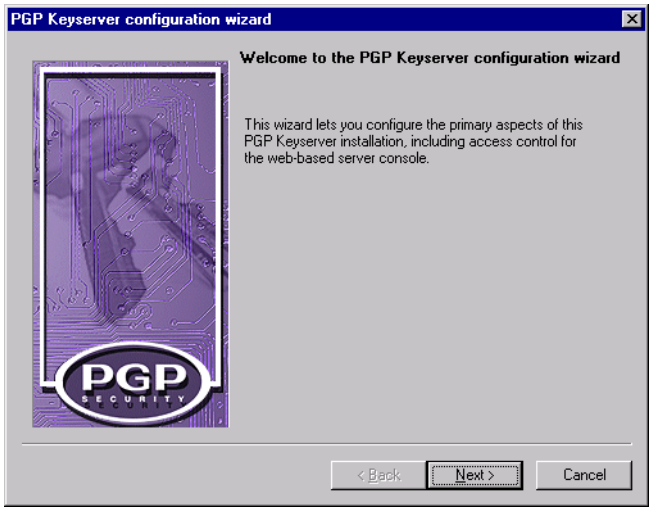
   The Readme.txt file appears listing the new features and other important information regarding PGP Keyserver, Enterprise Edition.

6. Review the Readme.txt file, then click **Next**.

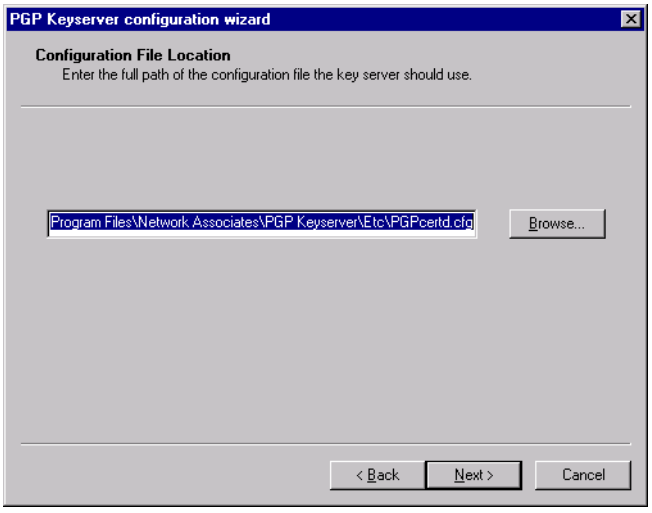
The **Choose Destination Location** dialog box appears.

7. Click **Browse** to navigate to a destination directory for your PGP files, then click **Next**. The **Start Copying Files** dialog box appears.
8. Review the installation settings, then click **Next**.

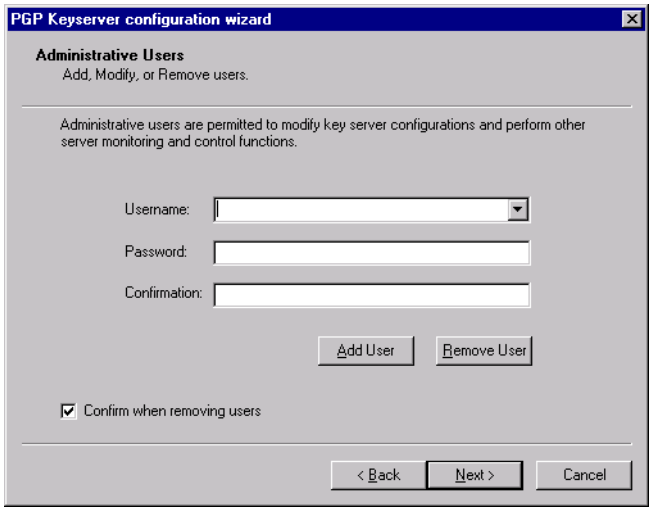
   The PGP Keyserver files are copied to the computer.
9. Click **OK** to complete the PGP Keyserver installation and start the PGP Keyserver configuration wizard.



10. Read the Welcome page and click **Next**. The configuration wizard displays the location of the configuration file.

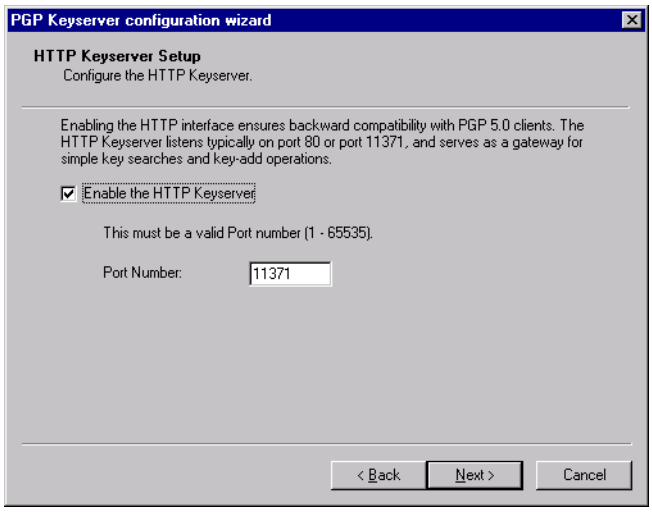


11. Click **Browse** to select a different configuration file, if desired. Click **Next**. The wizard displays the Administrative Users page.



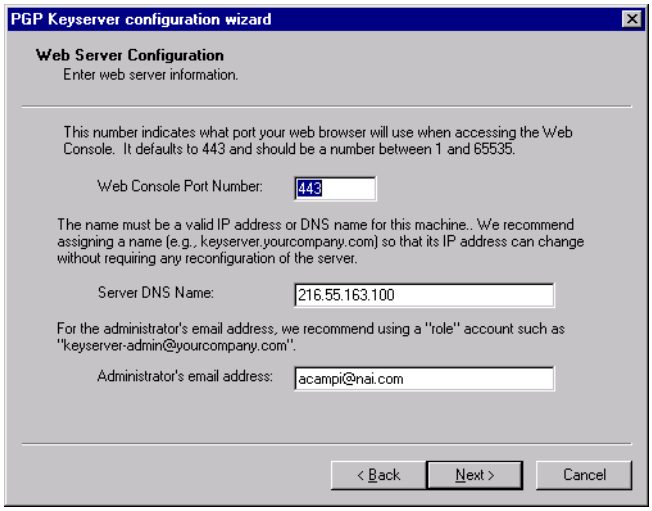
12. Add at least one administrative user to your configuration. Enter a password for that user, and then enter the password a second time to confirm the password. Click **Add User**. Repeat this process for each administrative user, then click **Next**.

    The wizard displays the HTTP Keyserver Setup page. This step ensures backward compatibility for PGP 5.0 clients.



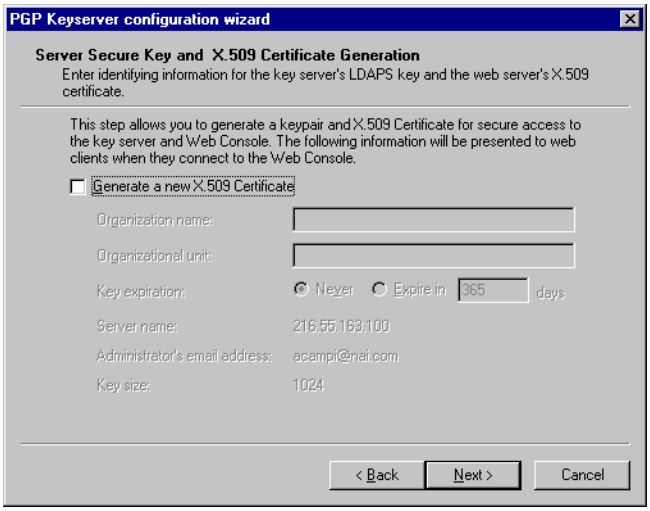
13. Verify that the wizard displays the desired port number. If it does not, enter the correct one. Click **Next**.

The wizard displays the Web Server Configuration page.



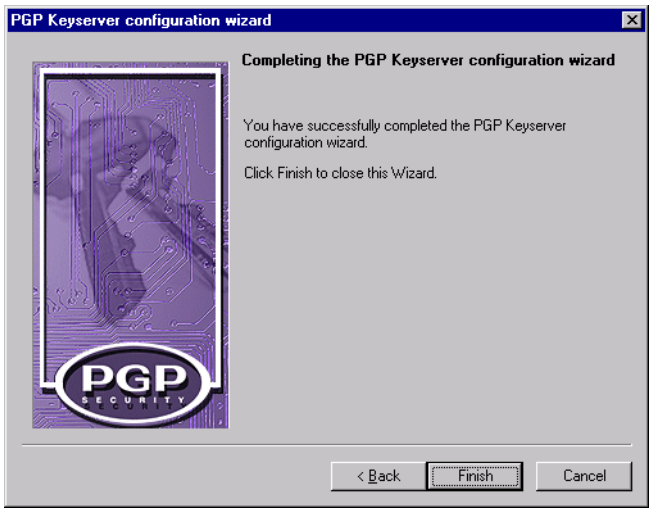
14. Verify that the information on this page is correct. The **Server DNS Name** must be a valid IP address or DNS name. Make any corrections required, then click **Next**.

    The wizard displays the Server Secure Key and X.509 Certificate Generation page.



    This panel allows you to generate a keypair and X.509 certificate for secure access to the PGP Keyserver. The information displayed on this panel is presented to Web clients when they connect to PGP Keyserver's Web Console. This key is also used as the default for the **ServerSecureKeyID** configuration setting. The configuration files for PGP Keyserver and PGPapache will be automatically updated to use this new key. Enter the required information.

15. Click **Next**. The wizard displays the Completing the PGP Keyserver configuration wizard page.



16. Click **Finish**. To update the PGP Keyserver with the configuration wizard information,
17. Click **Finish**. You can elect to use a passphrase to encrypt the Keyserver's private key. The paragraphs that follow describe when to use a passphrase and when to use a blank passphrase.

    Using a passphrase. If you use a passphrase, you must enter it each time the Keyserver starts PGPapache and whenever you use the **SecureMode** Required and **SecureMode** Optional settings in your Keyserver configuration. For more information, see **SecureMode** in the *PGP Keyserver Administrator's Guide*.

    Using a blank passphrase. If you require unattended operation of the Keyserver machine, leave the passphrase blank.

Installing the PGP Keyserver on a UNIX server

You can download the PGP Keyserver software from an authorized Web site or load the software from a CD-ROM. The file automatically extracts and installs all of the necessary software components in their proper directory locations. After you install the software, run the PGP Keyserver Configuration wizard to customize the PGP Keyserver to meet the needs of your site.

To install the Keyserver on a Sun SparcStation:

Note: To install the software, you must have root privileges.

1. Start the UNIX server.
2. Insert the PGP CD-ROM into the CD-ROM drive.
3. Navigate to the directory where the PGPkeyserver_x.x.x_Solaris file is located, and then begin installing the package by issuing the following command:

   pkgadd -d PGPKeyserv_7.0.0_Solaris.pkg

Note: If you install from a CD-ROM drive under Sun Solaris, you may receive a warning that tells you that the file system does not conform to ISO-9660 specifications. This is because the name of the file has more than eight characters. Ignore this warning; the install will proceed without problems.

4. Review the license agreement information, then type **Y** to accept the licensing terms.
5. The Web server bundled with PGP Keyserver provides a secure Web Console and HTTP keyserver interface for PGP 5.0 compatibility. Type **Y** to use this Web server.

   If you want to configure the Web server already installed on this system to serve the Web Console, type **N** and continue with Step 10 .
6. Type **Y** if you want the Keyserver to start after installation and automatically start after restarting the computer.
7. Type **Y** if you want the Web Console services to start after installation and automatically start after restarting the computer.

   The Solaris package Manager warns that the Keyserver binaries pgpcertd and pgprepd will be installed setuid. This ensures that the services will run as the unprivileged user named 'pgpkserv'. This prompt is not displayed if you answered N in Step 5 on page 27.
8. Type **Y** to allow the package installer to correctly install these files.

9. Type **Y** to complete the installation and run the post-install script (this runs in the background).

   The installer processes the package and system information, verifies disk space requirements, checks for conflicts with other packages, and then looks for setuid and setgid programs.

Note: The post-install script is executed with super-user permission during the process of installing this package.

   The post-install script then displays these instructions:

   Use the following command to run the Keyserver's configuration wizard and complete the configuration of the Keyserver's Web Console:

   cd /opt/PGPkeysrv/web/ ; ./config-wiz.pl

10. The following steps include instructions for each configuration wizard screen. Type **N** to proceed to the next page.

    a. Read the welcome screen and press the **Enter** key.

    b. Add at least one administrative user to your configuration. To do so, enter a username and password (enter the password a second time to confirm it).

    c. Enter the required information: **Server DNS Name**, **Web Server Port**, and **Administrator's E-Mail Address**. The server name must be a valid IP address or DNS name.

    d. To ensure backward compatibility for PGP 5.0 clients, PGP Keyserver includes an HTTP Keyserver feature. Verify that you want to provide this service and that the wizard is displaying the correct port number.

    e. Follow the instructions to generate a key and X.509 certificate.

       You can elect to use a passphrase to encrypt the Keyserver's private key. The paragraphs that follow describe when to use a passphrase and when to use a blank passphrase.

       Using a passphrase. If you use a passphrase, you must enter it each time the Keyserver starts PGPapache and whenever you use the **SecureMode** Required and **SecureMode** Optional settings in your Keyserver configuration. For more information, see **SecureMode** in the *PGP Keyserver Administrator's Guide*.

       Using a blank passphrase. If you require unattended operation of the Keyserver machine, leave the passphrase blank.

    f. Enter **Y** to start the Web server.

11. To verify that the product was installed properly, enter the following command:

    pkginfo -l pgpkeysrv

The status for the selected package should be "Completely Installed."

To upgrade the Keyserver on a Sun SparcStation:

1. Use `ps -ad | grep pgpcertd` and `kill` to stop PGPcertd:
2. Install the PGP Keyserver as you would for a brand new installation. For more information, see the instructions in the previous section, "Installing the PGP Keyserver on a UNIX server" on page 27.
3. Export the existing database.
   a. Change to the /opt/PGPkeysrv/bin directory.
   b. Use the PGPexport utility to export the database:

      ./pgpexport ../data /opt/dump.pgp
4. Configure the PGP Keyserver as described in the documentation and restart the program to institute the updated policies.
5. Re-import the keys from the old database.
   a. Change to the /opt/PGPkeysrv/bin directory.
   b. Use the PGPimport utility to re-import the database:

      ./pgpimport /opt/dump.pgp ldap://localhost
   c. Re-disable any keys that were disabled in the old installation.

3 Getting Started

This chapter outlines the steps required to configure the PGP Keyserver, Replication Engine, Web Console, and slave PGP Keyservers (optional).

Installation Overview

Note: You must have administrative privileges on the machine where you are installing the PGP Keyserver.

The software has three components: the PGP Keyserver, the Replication Engine (Engine), and the Web Console.

- The P*GP Keyserver* allows users to submit and retrieve keys from a database. The PGP Keyserver uses a set of user-defined policies to control key submission and retrieval.

  If your installation is large or if your users are in a number of different locations, a single PGP Keyserver may not meet your users demand for keys. As a result, you may require a PGP Keyserver on a number of systems:

  – *Master PGP Keyservers* perform all PGP Keyserver functions. A PGP Keyserver, replication log (a log where new and modified keys and destination slave PGP Keyservers are recorded), and a Replication Engine reside on each master PGP Keyserver.

  – *Slave PGP Keyservers* provide search functions.

  You can add slave or master PGP Keyservers to the same physical location or remote locations. Multiple master PGP Keyservers (PGP Keyservers that can perform all PGP Keyserver functions), are considered peers.

  For a complete description of PGP Keyserver configurations, see Chapter 7, The Replication Process.

- The *Replication Engine* allows you to replicate database changes on a master PGP Keyserver to other PGP Keyservers. The databases on the slave PGP Keyservers are automatically updated to reflect the contents of the database on the master PGP Keyserver.

  Installation of the Replication Engine is only required if you install multiple PGP Keyservers. For a complete description of the replication process, see Chapter 7, The Replication Process.

- The Web Console allows you to control and monitor the PGP Keyserver and Replication Engine via a Web browser. The Web Console also includes useful key management and monitoring features.

If you install one PGP Keyserver, all components (PGP Keyserver, Engine, and Web Console) are installed on the same machine.

Installing the PGP Keyserver, Replication Engine, and Web Console

After you install the PGP Keyserver, run the PGP Keyserver configuration wizard. On the Windows NT platform, the wizard starts automatically. Use the wizard to configure the following items:

- The PGP Keyserver configuration file that you want to use (Windows only).
- Identify Administrative users who can access the Web Console, modify PGP Keyserver configurations, and perform other PGP Keyserver monitoring and control functions.
- Enable the HTTP interface and identify the HTTP port number (this setup ensures backward compatibility for PGP 5.0 clients).
- Identify the Web server's port number, DNS name, and the Administrator's email address. (These values can be changed at any time.)
- Create a keypair and X.509 certificate that the PGP Keyserver presents to users to create a secure encrypted connection.

Displaying the PGP Keyserver's Web Console

The PGP Keyserver's Web Console gives you easy access to all Keyserver and Engine functions. To display the Web Console, enter the following URL in the location field of any Web browser:

https://<*hostname or IP address*>[:<*port*>]/keyserver/

Use the hostname (Server DNS Name) and port (Web Console Port Number) that you entered when you ran the PGP Keyserver's configuration wizard. By default, the port is 443. To verify this information, you can run the configuration wizard again from the **Start** menu (**Start—>Programs—>PGP Keyserver—>Configuration Wizard**). Click **Next** until you view the Web Server Configuration panel.

Note: For easy access, bookmark the PGP Keyserver page in your Web browser.

Windows: Note that you can also start the Web Console from the **Start** menu on machines where the PGP Keyserver is installed (**Start—>Programs—>PGP Keyserver—>Web Console**).

UNIX: The UNIX configuration wizard displays the URL.

Adding an additional PGP Keyserver (optional)

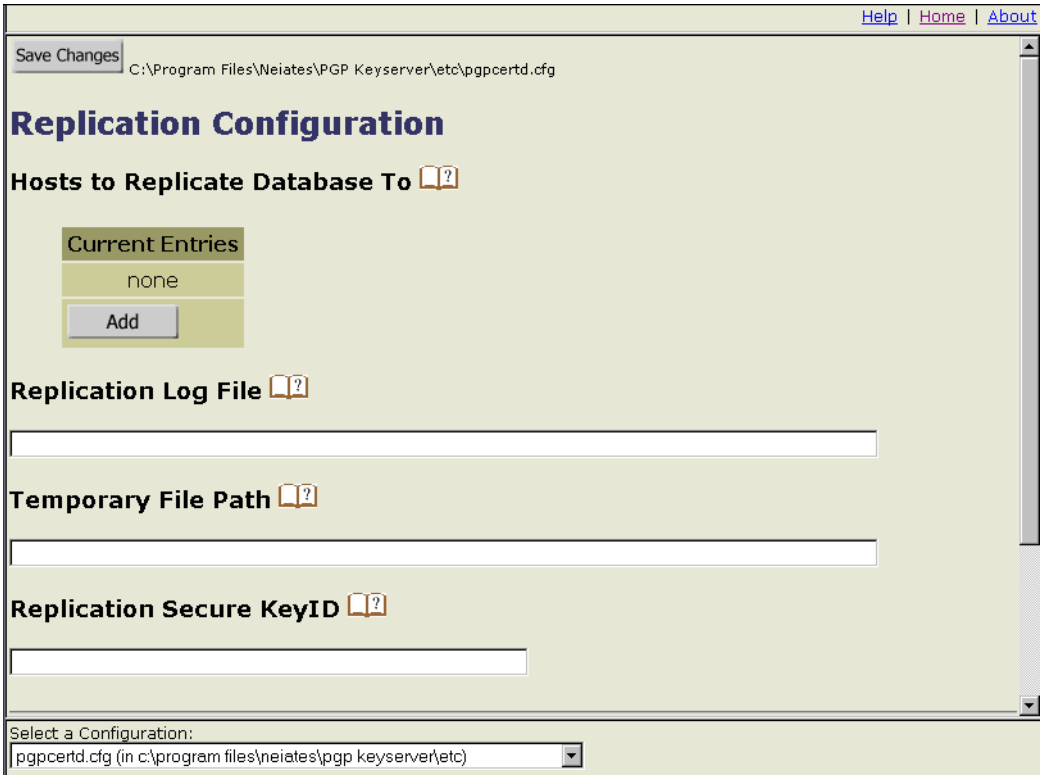
To add a new PGP Keyserver to an existing PGP Keyserver installation, follow the instructions in this section. They describe how to copy keys from the master PGP Keyserver to one or more slave PGP Keyservers.

To add an additional Keyserver to your configuration:

1. Install the PGP Keyserver software on the system that will run the new PGP Keyserver (see "Installing the PGP Keyserver, Replication Engine, and Web Console" on page 32).
2. Configure the PGP Keyserver software to send replication entries. Perform this step on any Keyserver machines that will accept updates from clients (this includes master Keyservers). Slave Keyservers need not complete this step. For more information about replication configurations, see Chapter 7, The Replication Process.
   a. On the Web Console click **Replication**.
   b. To identify the slave PGP Keyserver, click **Add** (**Hosts to Replicate Database To**) and enter the hostname or IP address of the new Keyserver).
   c. Click **Add**.
   d. Add a replication log file (**Replication Log File**). Enter the full pathname, for example:

C:\Program Files\PGP Corporation\PGP Keyserver\Logs\RepLog

e. Click **Save Changes**. The Web Console tells you it has updated the configuration file (Complete), or displays an error message.



3. If the changes will not be replicated over LDAPS, go to Step 4.

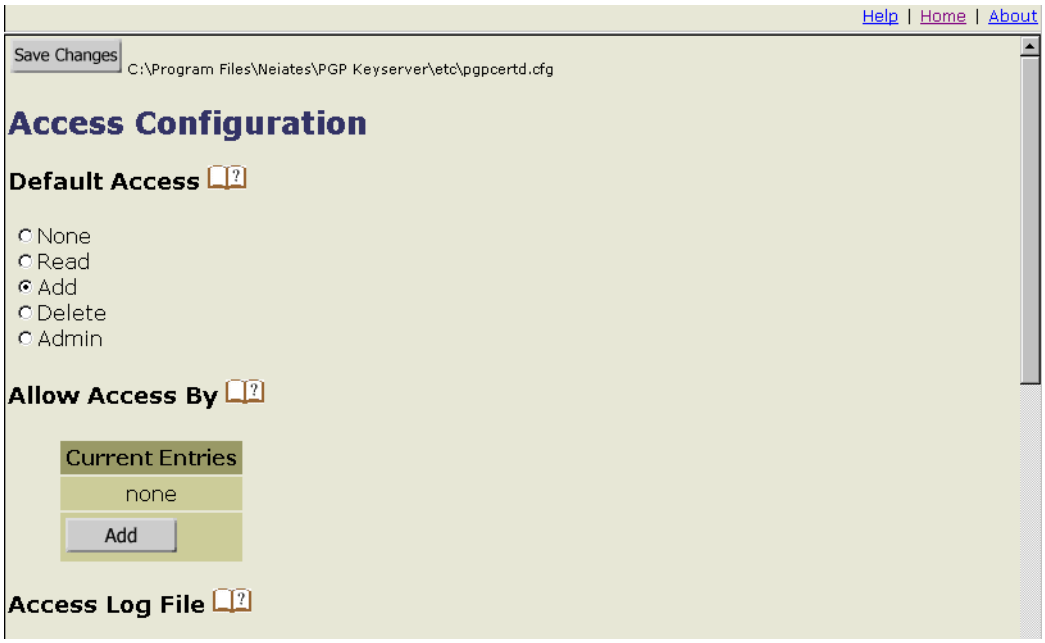
   If changes are replicated over LDAPS, the receiving PGP Keyserver must be configured to allow deletes from the sending PGP Keyserver's LDAPS key. For example, if Keyserver A is replicating over LDAPS to Keyserver B, and Keyserver A is using Key A as its server-side LDAPS key, Keyserver B must be configured to allow deletes from Key A.

   To configure Keyserver B, do the following:

   a. Click **Access** on Keyserver B's Web Console.

   b. On **Allow Access By**, click **Add**.

   c. Click **Host**, enter the hostname of Keyserver A's (**Identified as**), and select **Delete** for the level of access allowed (**Allow...**).



d. Click **KeyID**, enter the keyID for Keyserver A's LDAPS key (**Identified as**), and select **Delete**. If this is a new install, the LDAPS key is the only key on Keyserver A's keyring. To find out how to identify a key's KeyID, see see "Extracting key IDs: The PGP Key ID utility" on page 38.

e. Click **Save Changes**. The Web Console tells you it has updated the configuration file (Complete), or displays an error message describing the problem it found in the configuration file.

In addition, Keyserver A must be set up to allow users to delete their own keys or to allow an administrator to delete keys. Note that allowing the same administrator to delete keys on Keyserver B does not affect the replication process.

To configure Keyserver A, do the following:

a. Click **Access** on Keyserver A's Web Console.

b. On **Allow Access By**, click **Add**.

c. Click **Host**, enter the hostname for Keyserver B (**Identified as**), and click **Delete**.

d. Click **KeyID**, enter the keyID for Keyserver A's LDAPS key (**Identified as**), and click **Delete**.

e. Click **Save Changes**. The Web Console tells you it has updated the configuration file (Complete), or displays an error message.

4. Replicate the existing database to the new PGP Keyserver.

Before you use the **pgpexport** command, stop Keyserver A or configure it for Read Only mode.

a. Login to the machine where the existing PGP Keyserver resides (Keyserver A).

b. Run **PGP Export** (**Start—>Programs—>PGP Keyserver—>PGP Export**). For complete instructions for this utility, see "Exporting keys from the PGP Keyserver: The PGP Export utility" on page 39.

   Supply the pathname of the directory that contains the current database (for example, in Windows NT, C:\Program Files\PGP Corporation\PGP Keyserver\data; in UNIX, /opt/PGPkeysrv/data). **PGP Export** creates a file with the exported keys.

   If you are exporting a large database, you may want to use **PGP Export**'s **OutFile** and **KeysPerFile** arguments to write the database into multiple files. For details, see "Exporting keys from the PGP Keyserver: The PGP Export utility" on page 39.

c. Run **PGP Import** (**Start—>Programs—>PGP Keyserver—>PGP Import**) on Keyserver A, sending the exported keys to the new Keyserver on ldap://<Keyserver B>.

   For information on **PGP Import**, see "Importing keys to the PGP Keyserver: The PGP Import utility" on page 41.

Keyserver configuration

This section introduces you to a few features that you may want to configure before you begin using the PGP Keyserver.

Secure Mode

The PGP Keyserver includes a Secure Mode that you can use to perform deletions and other administrative tasks. When the PGP Keyserver is in Secure Mode, the PGP Keyserver cannot start unless it can successfully provide secure access by Transport Layer Security (TLS). TLS is a protocol based on SSL that provides encrypted and authenticated communications.

For more information about Secure Mode, see "Secure mode configuration settings" on page 83.

Controlling access to administrative operations

When the PGP Keyserver is running, the PGP Keyserver responds to client requests and requires very little attention. PGP allows you to submit (add) and retrieve (read) keys from the PGP Keyserver. In addition to submitting and retrieving keys, you can also delete keys from the PGP Keyserver.

To prevent unauthorized users from performing these operations, access is restricted to users with the proper authority. Access is regulated in the following ways:

- Password authentication
- Secure access via LDAPS (LDAP over TLS)
- Signature authentication (user submits a request, and the software checks the user's signature to make sure the user has permission to perform the operation)

Use the **Allow Access By** configuration setting (**Access** panel under **Server Configuration**, Web Console) to control user access to the read, add, delete, and administrative features (for details, see "Allow Access By - Allow" on page 86).

Controlling how rejected keys are handled

When a key does not pass policy, the key is rejected and a copy of the key is sent to the pending bucket. As System Administrator, you must periodically review the keys in the pending bucket. If the keys are valid, you can sign them and resubmit them to the PGP Keyserver. If the keys are invalid, you can delete them.

Use the **Action on Key Policy Failure** configuration setting (**Policy** panel under **Server Configuration**, Web Console) to control what happens to rejected keys. They can be sent to the pending bucket or ignored; when they are ignored, the PGP Keyserver generates an error message. For details see "Action on Key Policy Failure - PolicyFailures pending | error" on page 91.

When a key submitted to the PGP Keyserver passes policy, and that key already exists in the pending bucket, the key in the pending bucket is automatically deleted.

For more information on the pending bucket, see "Resolving keys in the pending bucket" on page 63.

Troubleshooting

If you experience problems:

1. Use the Windows Event Viewer or UNIX syslog to help identify what is wrong.
   - The Windows Event Viewer is available from the Start menu (**Start—>Programs—>Administrative Tools (Common)—>Event Viewer**). Select **Log—>Application** from the menu bar.

When you display the Event Viewer, make sure you are viewing the Application Log. Select **Log Settings** from the Event Viewer's **Log** menu. Set **Event Log Wrapping** to **Overwrite Events as Needed**. This will ensure that you see the most recent events generated by the Keyserver.

Use the Event Viewer's help to learn about the information displayed in each column of the Viewer.

– The name and location of the UNIX system log file is based on your system log configuration. For example, if you placed the line user.err /var/adm/user.log in the /etc/syslog.conf file, the Keyserver details for the user data are logged to this file. For more detailed information on system log configuration, consult the UNIX man pages.

Each entry displayed in the Windows Event Viewer and UNIX syslog describes an event that happened on the local machine. All entries in these logs that are associated with the PGP Keyserver have a source of "PGP Keyserver". This distinguishes these messages from those generated by other processes. The level of PGP Keyserver information in the two logs is controlled by the **LogLevel** configuration setting.

2. If you are unable to start the Web Console using your Web browser, note that it requires a browser with 128-bit encryption. Ensure that you have completed the configuration wizard and that PGPapache is running. On Windows NT, you may want to restart the PGPapache service after rerunning the configuration wizard.
3. For a complete list and description of LDAP error messages, see Appendix A, LDAP Error Messages.

Extracting key IDs: The PGP Key ID utility

The **MustSigID**, **AllowSigID**, and **Allow keyID** command configuration settings require you to identify the 64-bit key IDs. The **pgpkeyid** utility parses a keyring or ascii-armor key file and extracts these IDs automatically.

Use the following commands:

- pgpkeyid [-e] -k <keyring>
- pgpkeyid [-e] -a <asciiarmor>

The -e option lists the key ID of the encryption portion of the DSS/Diffie-Hellman or v4 RSA key. If you do not use this switch, you receive the signing portion of the key (in the case of DSS/Diffie-Hellman, DSS).

The -k option parses the key IDs from a PGP keyring.

The -a option parses the key IDs from an ASCII-armored key file.

For example, to display the 64-bit key IDs for the keys on the PGP Keyserver's keyring, position yourself in the PGP Keyserver\etc directory (the default directory for the PGP Keyserver's keyring), and enter the following command:

..\bin\pgpkeyid -k pgpkeyserver-pubring.pkr

The following is an example of the command line used to list all of the encryption keys in a keyring file:

pgpkeyid -e -k *keyring.pgp* > keyring.new

Exporting keys from the PGP Keyserver: The PGP Export utility

Use the PGP Keyserver's export feature to export keys from one PGP Keyserver to another or to a backup device (on Windows NT, **Start—>Programs—>PGP Keyserver—>PGP Export**). To export the contents of the PGP Keyserver, log on to the machine where the database is located (to perform the export, you must have read access to the database).

Please note the following:

- **PGPexport** uses ascii-armor format by default.
- Disabled keys are not exported by default. If disabled keys are exported using the **-d** switch, the disabled attribute is stripped when the keys are exported. Any disabled keys can be identified by adding the **-v** switch. The keyIDs of all exported keys will be displayed on your screen (Windows NT) or displayed on the standard error device (stderr, UNIX), and any disabled keys will also have "(disabled)" appended to the keyID.
- If the **-2** switch is not used, any key reconstruction data that is exported is enclosed in begin/end tags:

  -----BEGIN PGP KEY RECONSTRUCTION BLOCK-----

  <reconstruction data>

  -----END PGP KEY RECONSTRUCTION BLOCK-----

The following is the export command:

pgpexport [-ivVl12d] <directory> [OutFile [KeysPerFile]]

Each PGPexport switch is described in the following table.

PGPexport switch	Description
-v (lowercase) -V (uppercase)	Verbose mode. Displays the program's copyright and version number. When this option is used, the program prints version information then exits.
-i	Instructs the export process to ignore any errors that are encountered during the exportation of the certificate from the database. This option is useful when you know that there are errors, but you still need to perform the export.
<*directory*>	Identifies the directory where the PGP Keyserver database files are located. If you do not explicitly specify a directory, the current directory location is assumed.
[*OutFile*]	Identifies the path and filename where the **pgpexport** command writes the exported keys. By default the exported output is sent to the standard output device. To save the keys to a file, you must either use the *OutFile* parameter or redirect the output.
-l	(lowercase L) **pgpexport** checks to see if the key database is already in use. If the database is in use, **pgpexport** aborts. To override this and have **pgpexport** run even if the database is in use, use the -l option.
-d	Allow export of disabled keys. By default, disabled keys are not exported. If used in conjunction with verbose mode (-v), the keyIDs of disabled keys are marked as disabled.

[*KeysPerFile*]	Identifies the maximum number of keys to write to each file named by the *OutFile* argument. When *KeysPerFile* is used, the *OutFile* argument identifies the base name to use for each file, and a numerical extension is added to each file that the **pgpexport** command creates.
	For example, if your database (stored in the *data/* directory) holds 9,510 keys, and you want to write 100 keys to each file, enter the following command:
	`pgpexport data export/keyring 100`
	pgpexport creates 96 files in the subdirectory export and names them *keyring.00* through *keyring.96*.
	If the database holds 9,510 keys and you want to write all of the keys to one file, enter the following command:
	`pgpexport data export/keyring.pgp`
	All 9,510 keys are written to the file *keyring.pgp* in the subdirectory export.

PGPexport switch	Description
-2	Forces pgpexport to use the binary format found in version 2.x of the PGP Keyserver. When the binary format is used, reconstruction data stored in the PGP Keyserver is not exported. The standard error (stderr) output will display a warning to that effect. The binary output files can be used to exchange data with other PGP Keyserver products (such as the MIT keyserver) or with older versions of the Keyserver. Note that -2 can be combined with -1 (one). This exports version 1.x compatible keys, encoded as binary instead of ASCII-armored.
-1	(one) Indicates that the exported keyblock should be PGP Keyserver version 1.x compatible. New features are removed from the keys before exporting the keys. Keys are output as ASCII-armored keyblocks. Note that -1 can be combined with -2. This exports version 1.x compatible keys, encoded as binary instead of ASCII-armored.

If you use **PGPexport** without the **-d** option, and there were disabled keys on the PGP Keyserver prior to exporting the keys, **PGPexport** does not export the disabled keys.

Using **PGPexport** with the **-d** option will remove the disabled attribute from any disabled keys; you must re-disable them after importing to a fresh Keyserver database.

Importing keys to the PGP Keyserver: The PGP Import utility

When you set up your PGP Keyserver, you may want to import keys from an existing keyring file. You can import keys from any machine that has add privileges and access to the PGP Keyserver (n Windows NT, select **Start—>Programs—>PGP Keyserver—>PGP Import**).

Use the following import command to send all keys in a file to a PGP Keyserver:

pgpimport [-Vd] <*file*> ldap://<*hostname*>[:<*port*>]

Where <*file*> is the name of the file that you want to send to a PGP Keyserver, <*hostname*> is the name of the target PGP Keyserver, and <*port*> is the optional LDAP port number for the target PGP Keyserver. The LDAP port number is required only if the machine does not use the default LDAP port of 389. Use the **-V** option to display the program's copyright and version number, and the **-d** option to delete the imported file after the import runs to completion. Note that when you use the **-V** option, the program prints version information and then exits.

Please note that you can use the PGP Keyserver's secure port to import keys. For more information, see "Using Secure Mode" on page 73.

Removing the software

Windows NT/2000

To remove PGP Keyserver, use the **Add/Remove Programs** function from the Windows Control Panel.

Note: The uninstall program does not delete all files from the PGP Keyserver directory (by default, the PGP Keyserver directory is C:\Program Files\PGP Corporation\PGP Keyserver). After you run uninstall, remove any unwanted files and directories.

UNIX

To remove PGP Keyserver, go to the directory where the PGP Keyserver software is installed and enter the following command:

pkgrm PGPkeysrv

Note: The Solaris package remover removes the PGP Keyserver software, but does not delete any files that the PGP Keyserver generated or modified. After you remove the PGP Keyserver software, remove any unwanted files and directories. If necessary, the user pgpkserv and any files owned by it can be removed.

Adding and Removing Administrative Users

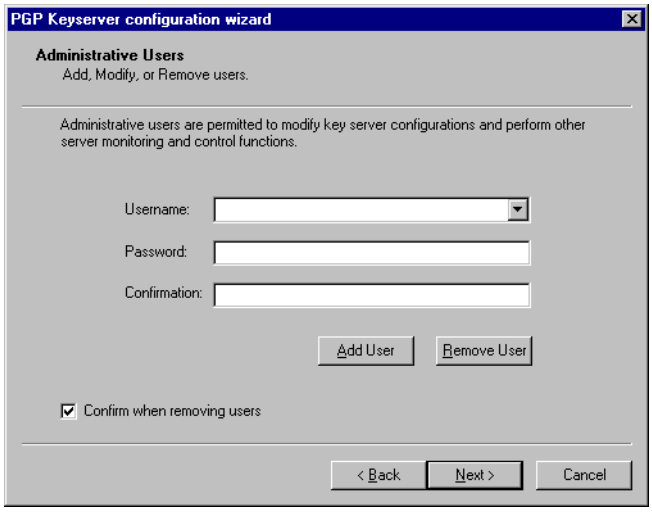
Use the Configuration Wizard to add and remove administrative users.

To add or remove administrative users:

1. Start the Configuration Wizard (**Start—>Programs—>PGP Keyserver—>Configuration Wizard**).
2. Click **Next** until the wizard displays the **Administrative Users** page.
3. To add a new administrative user, enter the user's name and password, and click **Add User**.

   To remove an existing user, select the user from the list and click **Remove User**. If **Confirm when removing users** is selected, confirm that you want to remove the user.

4. Click **Next** until you get to the wizard's last page—then click **Finish**. The Keyserver adds or removes the administrative user.



4 Controlling the Keyserver's Components

This chapter describes how to start and stop the PGP Keyserver, the Keyserver's Web Console, and the Replication Engine.

Starting the Web Console

You can start the Web Console from your browser. On Windows NT, you can also start the Web Console from the **Start** menu.

Starting the Web Console from a browser: UNIX and Windows

To display the Web Console, enter the following URL in the location field of any Web browser:

https://<*hostname or IP address*>[:<*port*>]/keyserver/

Use the hostname (Server DNS Name) and port (Web Console Port Number) that you entered when you ran the PGP Keyserver's configuration wizard. By default, the port is 443. To verify this information, you can run the configuration wizard again from the **Start** menu (**Start—>Programs—>PGP Keyserver—>Configuration Wizard**). Click **Next** until you view the Web Server Configuration panel.

Note: For easy access, bookmark the PGP Keyserver page in your Web browser.

Windows: You can also start the Web Console from the **Start** menu on machines where the PGP Keyserver is installed (**Start—>Programs—>PGP Keyserver—>Web Console**).

UNIX: The UNIX configuration wizard displays the URL.

Starting the Web Console from the Start menu: Windows

To start the PGP Keyserver Web Console from the **Start** menu on a system where the PGP Keyserver is installed:

1. Choose **Start—>Programs—>PGP Keyserver—>Web Console**. If the browser is unable to locate the Web Console, enter the URL in the browser's address field (for more information, see "Starting the Web Console from a browser: UNIX and Windows NT" on page 45).

Starting the PGP Keyserver

The PGP Keyserver starts automatically each time you reboot your system.

Restarting the PGP Keyserver

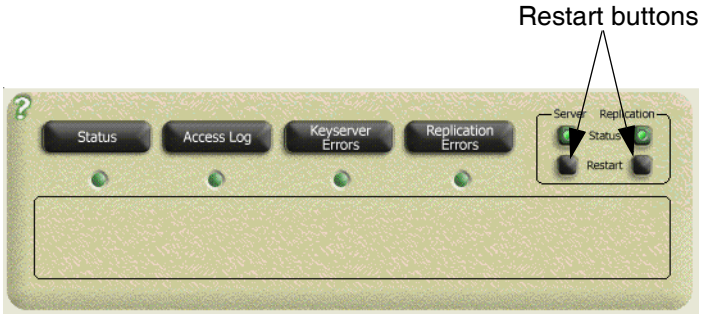
You can restart the PGP Keyserver from the Web Console or the command line.

Restarting the PGP Keyserver from the Web Console

The PGP Keyserver's Web Console includes start and restart features.

To start or restart the PGP Keyserver from the Web Console:

1. Start the Web Console.
2. Click **Server Control** under **Console** (Web Console's left panel).
3. Click **Restart** (upper right corner on console).



Starting the PGP Keyserver from the command line

Windows

When you start the PGP Keyserver from the command line, you specify the PGP Keyserver's run-time options. You must use the **-g** command line switch.

The following is the command with the appropriate command line switches:

pgpcertd [-a] [-c] -g [-n] [-s] [-t <*port*>] [-f <*file*>] [-p <*port*>] [-d <*level*>]

Each command line switch is described in the table on page 48.

When you start PGP Keyserver from the command line, you must also start PGPapache from the command line in order for the Web Console to properly detect the PGP Keyserver. If one is running as a service, the other must also run as a service in order for the two to communicate.

Use the Windows Services Control Panel, available from the Windows Start menu, to stop the PGPapache service. Then use one of the following commands to restart the service:

PGPapache.exe -f "c:\<path to config file>"

or

PGPapache.exe -d "c:\<path to install directory\web>"

UNIX

To start the PGP Keyserver, go to the directory where the PGP Keyserver software binary files are installed (/opt/PGPkeysrv/bin) and enter the following command with the appropriate command line switches:

pgpcertd [-a] [-V] [-c] [-s] [-t <*port*>] [-f <*file*>] [-p <*port*>] [-d <*level*>]

Keyserver command line switches

Keyserver Command Line Switch	Description
-a	Instructs the PGP Keyserver to assume that all signatures have already passed the policy requirements. All other policy checks are enforced. Use this option to copy a large number of verified keys from one PGP Keyserver to another.
-c	Checks the current configuration file for accuracy. Does not start the PGP Keyserver. When you make configuration changes, use this option to verify that the new configuration values are valid.
-f <*file*>	Identifies the configuration file that the PGP Keyserver uses. If -f is not specified, uses default found in the Windows registry (configured with the configuration wizard) or ../etc/pgpcertd.conf or .cfg.
-g	Windows. Tells the software to start as a command line PGP Keyserver rather than an NT service. Required if the PGP Keyserver is invoked on the command line.
-p <*port*>	Identifies the port number that the PGP Keyserver listens to for client requests and certificate submittals. Defaults to the port number listed in the configuration file or port 389. This option can also accept -1 (minus one), as long as **SecureMode** is correctly configured; regular LDAP connections will be disabled.
-V	Displays the program's copyright and version number. When this option is used, the program prints version information then exits.

Keyserver Command Line Switch	Description
-d <*level*>	Turns on the debug mode and provides a level of information based on the level you select. The following are the debug levels: Debug Level — Description 1 — Prevent process from detaching from terminal (UNIX) 2 — All normally logged event messages, as if 'LogLevel verbose' was set. 4 — Function trace 8 — LDAP arguments 16 — Shows active threads 32 — LDAP filter display 64 — Configuration file statistics 128 — ACL statistics 256 — LDIF parsing errors 512 — More detailed error conditions 1024 — More detailed processing messages 2048 — Info conditions 8192 — Misc messages 65535 — All of the above debug levels combined. NOTE: This switch is primarily used for debugging purposes. Do not use this switch unless you are very familiar with this process or you are consulting with a Technical Support Engineer.
-s	When used in conjunction with **SecureMode** set to Optional, allows you to run the PGP Keyserver automatically from a script (PGP Keyserver will start unattended with LDAPS disabled). If you do not use the **-s** option and LDAPS is enabled, the PGP Keyserver will prompt for a passphrase, if needed.
-t <*port*>	Identifies the port number that the PGP Keyserver listens to for Secure Mode. Defaults to port 636. If **-t** is not present, the PGP Keyserver uses the value for **SecurePort** found in the configuration file. **-t -1** (minus one) disables LDAPS and Secure Mode.

Restarting the PGP Keyserver when there are configuration errors: UNIX and Windows

If the PGP Keyserver is restarted using the Web Console (or, on UNIX, by sending it a SIGHUP), and there are errors in the configuration file, the PGP Keyserver will stop serving requests. This may occur if incorrect changes are manually applied to the configuration file.

- On UNIX, the daemon will exit. Correct the configuration errors and use the **Restart** button on the Web Console's **Server Control** panel to restart the Keyserver.

- On Windows, the Keyserver will be disabled (a blinking red light appears on the Web Console's **Server Control** panel). Correct the configuration errors and restart the Keyserver by stopping and starting the NT Service (Windows Services Control panel) or by clicking the blinking red light on the Web Console's Server Control panel.

To avoid this condition, use the Web Console to make configuration changes (the Web Console verifies your configuration changes automatically), or use the **-c** option to the pgpcertd executable to manually check for errors in a manually updated configuration file.

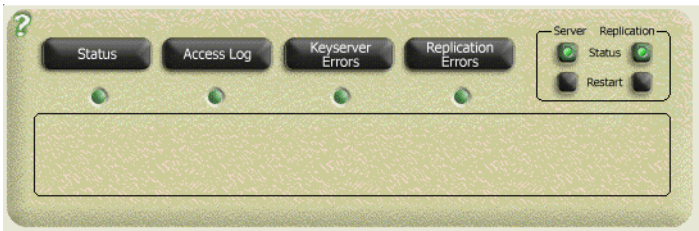
Running multiple PGP Keyservers on the same machine: UNIX and Windows

You can start any number of PGP Keyservers from the command line as long as each Keyserver uses a different set of port numbers, database directories, access log files, public keyring files, private keyring files, and replication log files. This is most effective in UNIX; for details, see "Starting the PGP Keyserver from the command line" on page 47.

Verifying that the PGP Keyserver is running

To verify that the PGP Keyserver is running, start the Web Console and click **Server Control** (Web Console's left panel). The color of the **Status** buttons (upper-right corner) tells you if the PGP Keyserver and Engine are running.

Color	Status
Green	Running. The PGP Keyserver is accepting both Web Console and client requests.
Yellow	Working. The PGP Keyserver is running but is momentarily unable to serve Web Console requests. It may be waiting for a reset operation to complete, which may take up to 300 seconds (or as specified in the **TimeLimit** configuration setting). For more information about the **TimeLimit** setting, see "Time Limit - TimeLimit <seconds>" on page 80.
Blinking Red	Disabled. The PGP Keyserver has been disabled by a Web Console.
Red	Not Running. The PGP Keyserver is not running.



If you prefer to use the standard UNIX facilities, use the following command to display the PGP Keyserver process and port:

```
ps -aef | grep pgpcertd
```

When the PGP Keyserver starts, it creates the following files:

- **/tmp/pgpcertd.pid**. Contains the PGP Keyserver's process ID (PID). If multiple PGP Keyservers are running on the same machine, this file contains the PID of the last PGP Keyserver started (HUP'd).
- **/tmp/pgpcertd.<*port*>.pid**. Where <*port*> is the port number for the PGP Keyserver. This file contains the PID of the PGP Keyserver running on the specified port. If one PGP Keyserver is running on the machine, this file is identical to the /tmp/pgpcertd.pid file.

If the PGP Keyserver is not running, examine the system log file (syslog) to find out what prevented the PGP Keyserver from starting.

For more information on how to configure and examine the system log file, "NT Event Log and system log file" on page 69.

One of the most common errors occurs when you try to start the PGP Keyserver when it is already running. When this happens, the PGP Keyserver may appear to work, but it does not apply configuration changes. A message appears in the system log file indicating that the PGP Keyserver could not bind to the specified port.

To find out if the PGP Keyserver is running, enter the following command:

```
netstat -a
```

Look under the TCP "Local Address" for a process running on the PGP Keyserver's port.

Stopping the PGP Keyserver

There are several ways to stop the PGP Keyserver under Windows:

- Use the Windows Services control panel to stop the PGP Keyserver.
- Enter the following at the command line:

  NET STOP "PGP Keyserver"

  The quotes are required.
- If the PGP Keyserver was started at the command line, press CTRL-C in the same console window to stop the PGP Keyserver.

To stop the PGP Keyserver under UNIX, use the SysV init script:

/etc/init.d/pgpkeyserver stop

Starting the Replication Engine

Windows: If configured to do so, the Engine starts automatically each time you reboot your system. The startup options are set in the Windows Services control panel.

UNIX: You can start the Replication Engine from the Web Console (**Server Control** panel). You can also symlink the SysV init script to perform this function:

ln -s /etc/init.d/pgpreplication /etc/rc2.d/S97pgpreplication

Restarting the Replication Engine

To restart the Engine from the Web Console, click **Restart** on the Web Console's **Server Control** panel.

Starting the Replication Engine from the command line

Windows

You must use the **-g** command line switch. PGPapache must also be run from the command line in order for the two to communicate.

When you use this method, the Engine runs with the options you enter on the command line, and the values displayed on the Web Console reflect those values:

pgprepd -g [-f <*file*>] [-t <*directory*>] [-r <*file*>] [-c] [-o] [-d<*level*>] [-s]

The table on page 54 describes the Replication Engine's command line switches.

Note: When you start the Replication Engine from the command line, you can run the Replication Engine in a debug mode useful for resolving problems. For details, see the table that follows.

UNIX

If you have installed the Engine, use the following command to start the Engine:

pgprepd [-V] [-f <*file*>] [-t <*directory*>] [-r <*file*>] [-o] [-c] [-d <*level*>]

Replication Engine command line switches.

Engine command line switches	Description
-f *<file>*	Identifies the configuration file that you want the Replication Engine to use. By default, the Replication Engine uses pgpcertd.cfg (in UNIX, pgpcertd.conf*)*, in the ../etc/ directory. This file contains all of the configuration settings that affect the Replication Engine.
-t *<directory>*	Identifies the directory for the Replication Engine's temporary files. This option overrides the TempPath configuration parameter. For more information, see "Temporary File Path - TempPath <path>" on page 93.
-r *<file>*	Identifies the log file you want the Replication Engine to use. This option overrides the RepLogFile value in the configuration file.
-o	During normal operation, the Replication Engine continuously monitors the replication log file for new entries. When you use this option, the Replication Engine looks at the replication log file only once. After processing all the entries, it then exits.
-c	Checks the current configuration file for accuracy. When you make configuration changes, use this option to verify that the new configuration values are valid. UNIX. Temporarily starts and stops the Replication Engine. Configuration warnings and error messages are sent to the standard error device (stderr). Windows. Does not start the Replication Engine.
-d *<level>*	Turns on debug mode and gives you information based on the level you choose. The levels are the same as those for the PGP Keyserver.
-g	Windows. Required if the Engine is invoked from the command line. Indicates that the Engine is not running as an NT service.
-V	Displays the program's copyright and version number. When this option is used, the program prints version information then exits.

The temporary files used by the Replication Engine can become quite large. Make sure they are stored on a partition that is large enough to hold this data. Use the **-t** option or the **TempPath** configuration setting to explicitly designate where these temporary files are stored.

Each time the Replication Engine starts it creates the /tmp/pgprepd.pid file which contains the process ID (PID). To verify if the Replication Engine is running, check to see if the process is running.

Note: If you shut down the Engine and want it to restart without continuing its replication from where it left off, remove the replica log file (RepLogFile setting) and the pgprepd related files from the temporary directory (TempPath setting).
If you shut down the Engine and want it to restart replication from the start of the RepLogFile, remove only the pgprepd related files from the temporary directory (TempPath setting). For more information, see "Temporary File Path - TempPath <path>" on page 93 and "Replication Log File - RepLogFile <filename>" on page 92.

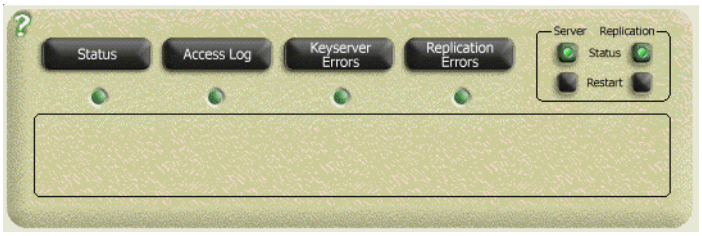
Engine authentication

The Engine's TLS key is selected from the keys in the Engine's key ring. The location of the key ring is defined in the configuration file. There are two ways to select the Engine's TLS key: define the **ReplicationSecureKeyID** setting in the Keyserver's configuration, or start the Engine from the command line and select the key when prompted.

- If the selected key requires a passphrase and the Engine is run as a command line application, the Engine prompts for a passphrase.
- If the selected key does not require a passphrase, the Engine does not prompt for one.

Verifying that the Replication Engine is running

You can view the Engine's status on the upper right corner of the Web Console's **Server Control** panel.



Running multiple Engines on the same machine

You can start any number of Engines from the command line as long as each Engine uses a different set of public keyring files, private keyring files, temporary file paths, and replication log files.

Stopping the Replication Engine

There are three ways to stop the Engine under Windows:

- If the Engine was started from the Web Console, enter the following at the command line:

  NET STOP "PGP Replication Engine"

  The quotes are required.
- If the Engine was started at the command line, press CTRL-C in the same console window to stop the Engine.
- Use the Windows Services control panel.

To stop the Engine under UNIX, use the SysV init script:

/etc/init.d/pgpreplication stop

5 Using the PGP Keyserver and Replication Engine

This chapter tells you how to use both the PGP Keyserver and the Replication Engine.

Using the Web Console

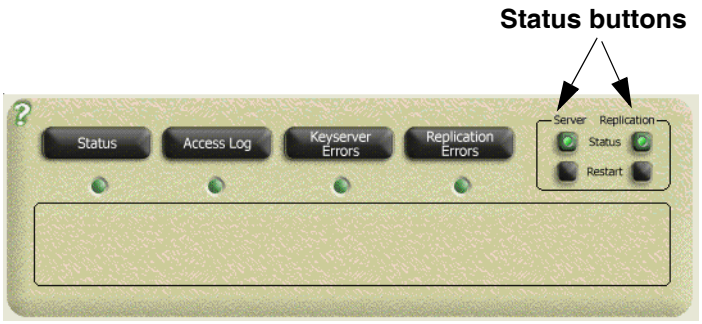
Server Configuration panels, Console panels, and documentation

Current configuration file

Reviewing PGP Keyserver and Engine status

Click **Server Control** (Web Console's left panel) to review PGP Keyserver and Engine status. The color of the **Status** buttons (upper-right corner) tells you if the PGP Keyserver and Engine are running.

Color	Description
Green	Running. The PGP Keyserver is accepting both Web Console and client requests.
Yellow	Working. The PGP Keyserver is running but is momentarily unable to serve Web Console requests. It may be waiting for a reset operation to complete, which may take up to 300 seconds (or as specified in the **TimeLimit** configuration parameter).
Blinking Red	Disabled. The PGP Keyserver has been disabled by a Web Console.
Red	Not running. The PGP Keyserver is not running.



To view Help for the **Server Control** panel, click the question mark in the top left corner of the panel.

Restarting the Keyserver or Engine

Use the **Restart** buttons on the **Server Control** panel to reset the Keyserver or Engine. This action causes the Keyserver or Engine to accept changes that you have made to the configuration file and reset statistics.

Note that if you change the following configuration settings via the Web Console, you receive a warning message that you must stop and restart the Windows NT service or UNIX daemon for the Keyserver to accept the changes:

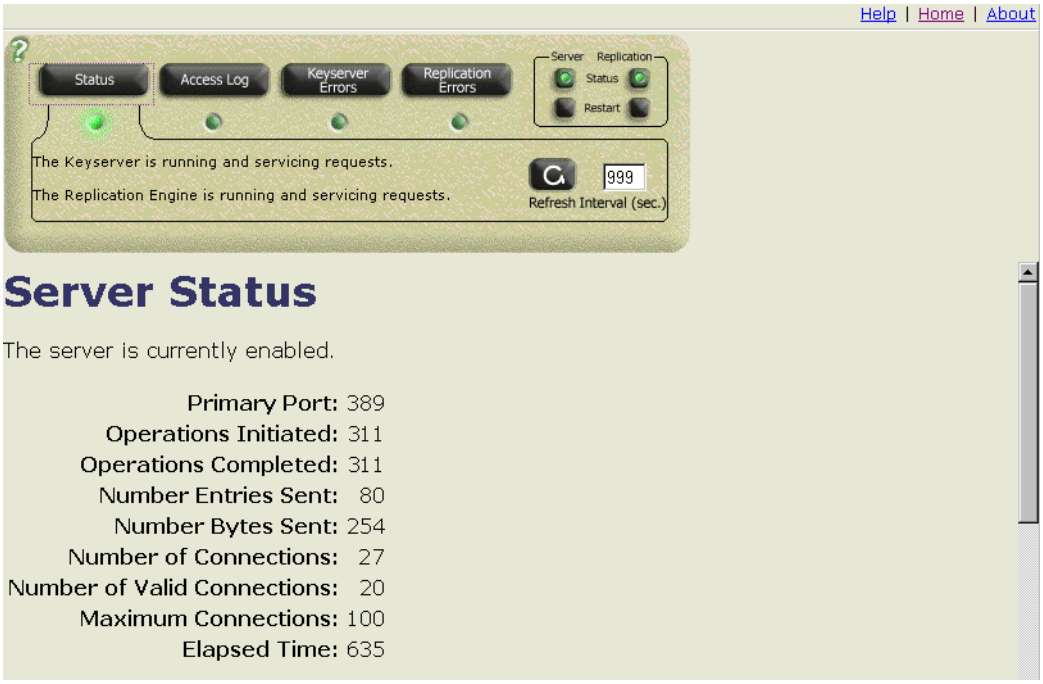
- ServerSecureKeyID
- ReplicationSecureKeyID
- PublicKeyRing
- PrivateKeyRing
- SecureMode

Detailed status information: The status report

Click **Status** (**Server Control**) for detailed information about PGP Keyserver and slave Keyserver status. The Server Status report gives information about the keys submitted and retrieved from the PGP Keyserver, and the Replication Engine Status report tells you how the replication is progressing on each of the slave machines where you are replicating information. Information includes the host machine and the port it is connected to.

When you click **Status** a **Refresh** button and box appear in the **Server Control** panel.

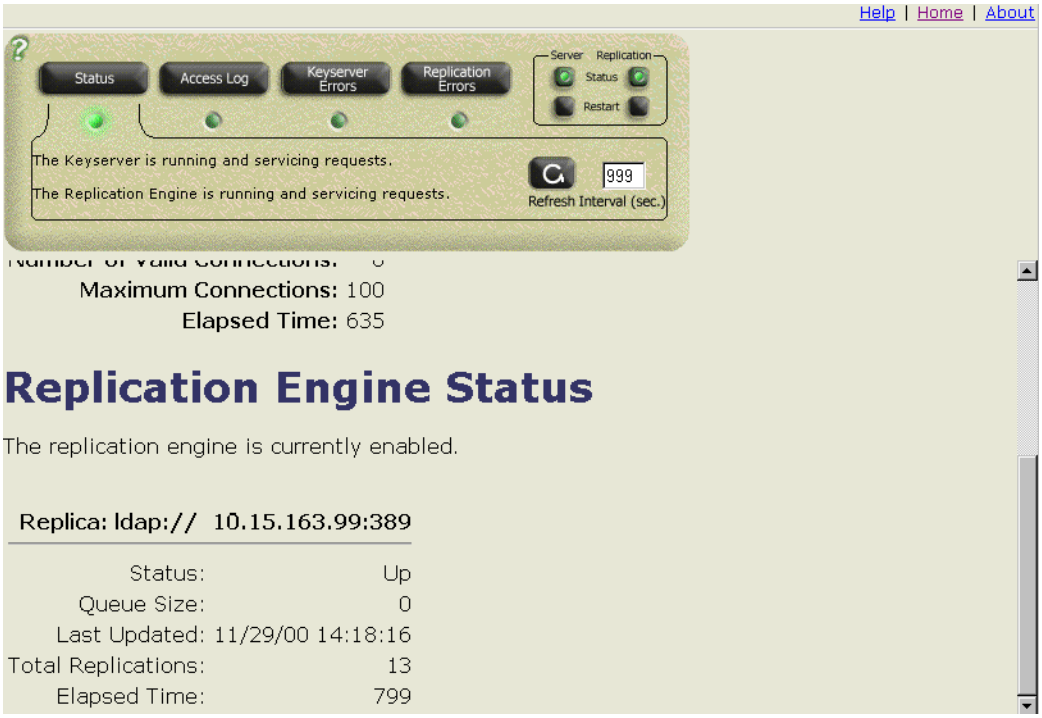
- To change how often the display is refreshed, enter a value in the box and click **Refresh**.
- To manually refresh the display, click **Refresh**.



The Web Console displays the following statistics for the PGP Keyserver:

- **Status**. Enabled (the PGP Keyserver is running and available to serve requests), Working (the PGP Keyserver is running but is momentarily unable to serve Web Console requests), Disabled (the PGP Keyserver has been disabled by a Web Console), or Not running (the PGP Keyserver is not running).
- **Primary Port**. The port that the PGP Keyserver is using.
- **Operations Initiated**. Number of client operations the PGP Keyserver has started processing since the PGP Keyserver has been started or reset, including those operations which are still in progress. Operations include searches, adds, and deletes.

- **Operations Completed**. Number of client operations the PGP Keyserver has processed since the PGP Keyserver was started. This number is based on the following operations: client connections opened, client connections closed, searches, and adds.
- **Number Entries Sent**. Total number of keys returned to clients through searches since the PGP Keyserver was started.
- **Number Bytes Sent**. Number of bytes transmitted between the PGP Keyserver and client(s) since the PGP Keyserver was started.
- **Number of Connections**. Total number of connections made to the PGP Keyserver since it was started. Under most circumstances, each connection is equivalent to one client session (for example, a client performing a search).
- **Number of Valid Connections**. Number of active connections on the PGP Keyserver. Each connection consists of multiple operations. The number is typically low since most client connections are generally short-lived.
- **Maximum Connections**. Maximum number of active connections allowed.
- **Elapsed Time**. Time since the PGP Keyserver was last started.



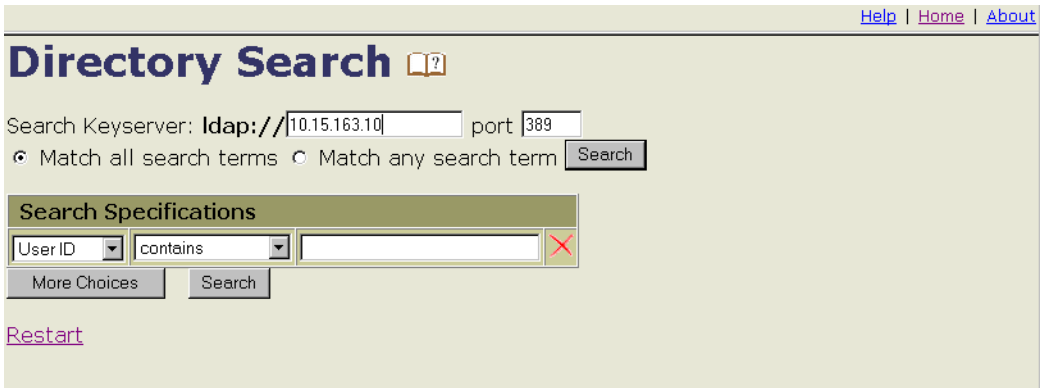
The console displays the following statistics for each replica or slave Keyserver, under Replication Engine Status:

- Status. Up (the PGP Keyserver is available), down (tried, not available), Untried (not yet tried).

- **Queue size**. Displays the number of replication operations that are currently lined up for processing.
- **Last updated**. Displays the time that the last successful replication occurred for the selected host.
- **Total replications**. Displays the total number of successful replications that have occurred since the Engine was last started.
- **Elapsed time**. Time since the Engine was last started.

Performing searches on the PGP Keyserver: Directory Search

Use this feature, similar to the PGPkeys search feature, to find a key or keys on a PGP Keyserver. You can search by userid, keyid, signerid, keytype, creation, expiration, status, and size, and, if desired, use the following secondary search parameters: is, is not, contains, and does not contain.



To search for a key:

1. Select **Directory Search** from the PGP Keyserver's Web Console.
2. Enter the name of the PGP Keyserver (defaults to the machine the Web Server is running on or the last used Keyserver).
3. Enter the PGP Keyserver's port (defaults to the current PGP Keyserver's port). At this point you can view all keys on the PGP Keyserver by clicking **Search**. To search for specific keys, go to Step 4.
4. Click your selection: **Match all search terms**, or **Match any search term**.
5. Select the primary search criteria from the first menu (**User ID, Key ID, Signer ID, Key Type, Key Status, and Key Size**).
6. If desired, select the secondary search criteria from the second menu—these vary, depending on the primary search specifications.
7. In the next box, enter the value you want the PGP Keyserver to search for, for example, `aBrown`.

8. If desired, click **More Choices**. The search program displays all available search specifications excluding the ones you have already selected. Select additional search specifications. Repeat this step as often as desired.
9. Click **Search**.
10. To restart your search, click **Restart**.

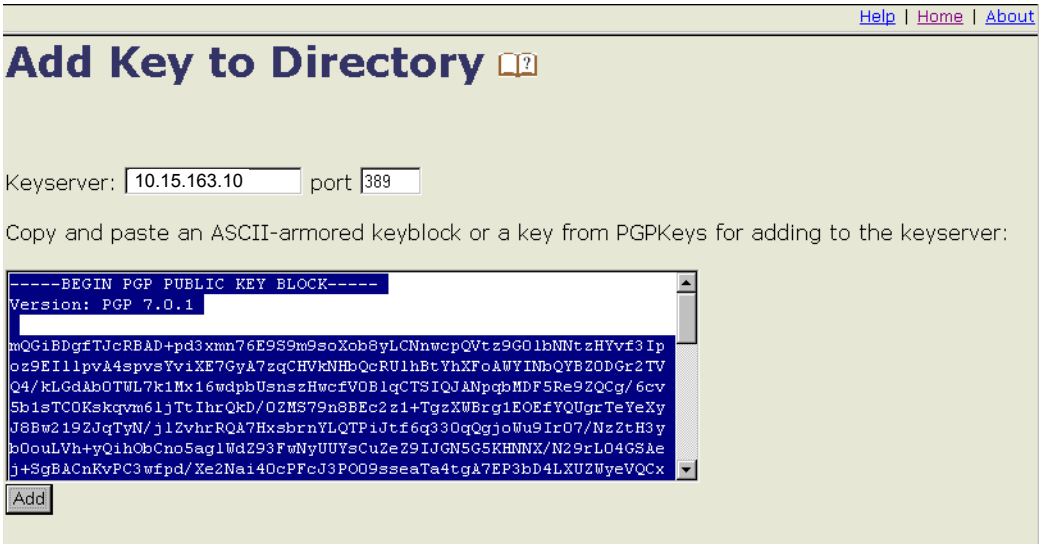
Adding keys to the PGP Keyserver

Use this feature to copy a key from PGPkeys to the PGP Keyserver.

Caution: To prevent anonymous users from adding keys to the Keyserver via the Web Console, use "Allow IP 127.0.0.1 none" and "Allow IP <Keyserver's real IP address> none". Using these settings forces users to use the PGP client to add keys. For more information about configuring access controls, see "Allow Access By - Allow" on page 86.

To copy a key from PGPkeys to the PGP Keyserver:

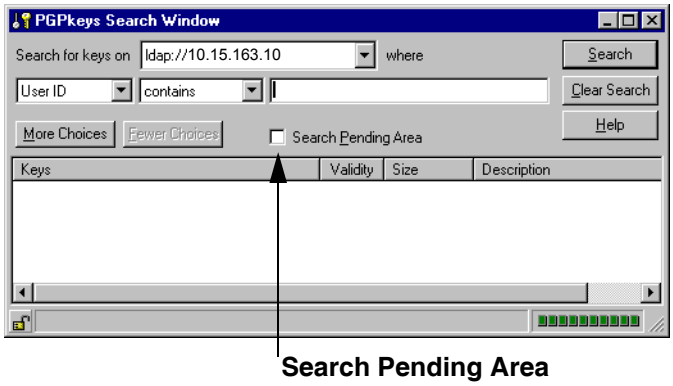
1. Select **Add Keys** from the PGP Keyserver's Web Console.
2. Enter the name of the PGP Keyserver (defaults to local PGP Keyserver or the last used Keyserver).
3. Enter the port of the PGP Keyserver (defaults to local PGP Keyserver's port or the last used port number).
4. Copy the target key from PGPkeys (select the key in PGPkeys, and select Copy from the **Edit** menu).
5. Put the insertion point in the box on the **Add Keys** panel, paste the key, and click the **Add** button.



Resolving keys in the pending bucket

As system administrator, it is your responsibility to identify the keys rejected due to policy infractions. The pending bucket is an area on the PGP Keyserver that holds keys rejected by the PGP Keyserver. To access the pending bucket, use the PGPkeys Search feature (**Server—>Search**), and click **Search Pending Area**.

For complete details on how to search for and manage keys, consult the *PGP Administrator's Guide*.



Viewing the Access log and error reports

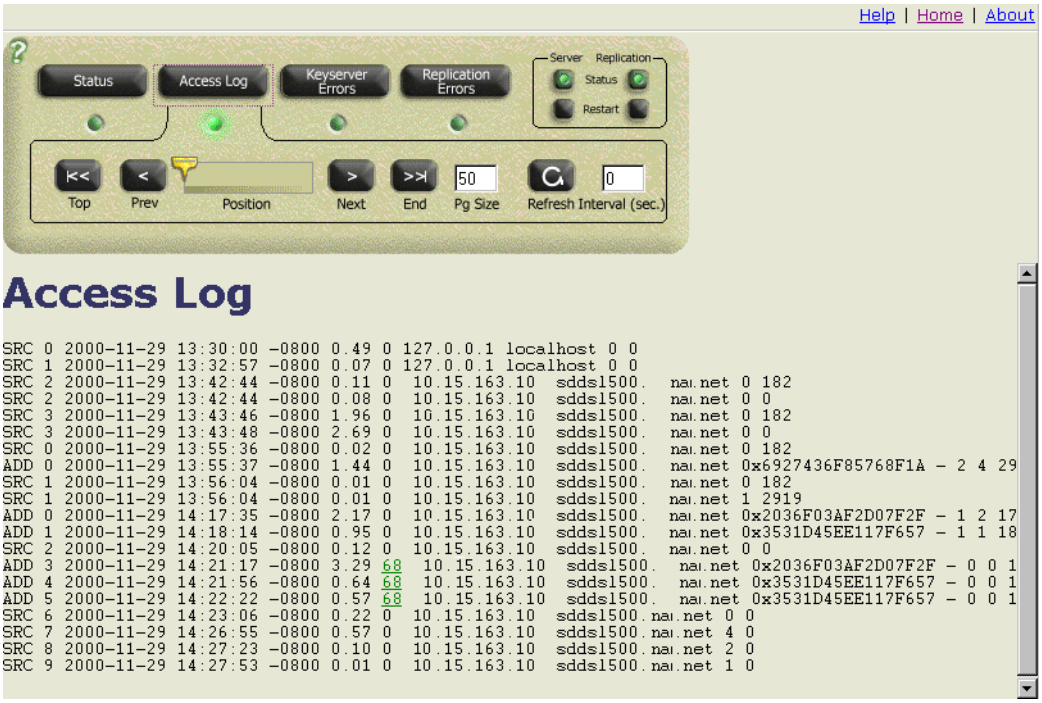
Use the **Access Log**, **Keyserver Errors**, and **Replication Errors** buttons on the **Server Control** panel to display the log files associated with client accesses to the PGP Keyserver (**Access Log**), error messages generated by the PGP Keyserver (**Keyserver Errors**), and error messages generated by the Replication Engine (**Replication Errors**).

Use the controls that appear on the **Server Control** panel to view the contents of a log (buttons are not displayed until you click a log button).

Use this button...	To do this...
Top	Displays first page of logged data (oldest data).
Prev	Displays the previous page of logged data.
Position	Displays a different portion of the log.
Next	Displays the next page of logged data.
End	Displays the last (newest) page of data.
Pg. Size	Displays the number of lines of logged data per page.
Refresh	Refreshes the status display. Use alone or in conjunction with the **Pg. Size** or **Interval**.
Interval	Displays how frequently the end of the log file is refreshed automatically. Use in conjunction with **End**.

View the Access Log

To view the Access Log, click **Server Control** on the Web Console's left panel, and click **Access Log**. The PGP Keyserver displays the Access Log below the controls.



The Access Log gives a chronological view of all the requests the PGP Keyserver has processed. You may also examine the Access Log file directly on the PGP Keyserver. The level of information is controlled by the PGP Keyserver's **AccessLogDetails** configuration setting. The format of the log file is controlled by the **AccessLogFormat** configuration setting:

- If **AccessLogFormat** is set to **Version1**, log entries appear in the following format:

  operation session time result IP host type-specific

- If **AccessLogFormat** is set to **Standard**, log entries appear in the following format:

operation session time duration result IP host [type-specific...]

Value in log	Description
operation	A three letter code describing the type of request submitted to the PGP Keyserver. The following codes may appear in the Access Log File: BND = Bind operation BDR = Key Reconstruction Attempt UBD = Unbind operation ABN = Abandon SRC = Search operation SRP = Search for Reconstruction Prompts SRD = Search for Reconstruction Data ADD = Add certificate operation MOD = Modify entry operation DLT = Delete operation DIS = Disable operation DAP = LDAP operation ??? = Unknown operation
session	A connection identifier that ties together multiple operations done over the same connection (except for reconstruction).
time	The time the request was generated. The time, given in the server's local time, is expressed using the following format: YYYY-MM-DD HH:MM:SS +/-TZTZ (where TZTZ is HHMM of the time zone).
duration	The length of time from when the request was first received until it is logged. This is in seconds with 100ths shown after the decimal point and a leading "0" (zero) if less than one second. This time is accurate to at least 1/100th of a second.
result	An LDAP result code in decimal format. For more information on the meaning of these codes, see Appendix A, LDAP Error Messages.
IP	The IP address, in dotted decimal format, of the client making the request.
host	The DNS hostname of the client making the request. If the address cannot be resolved to a domain name or if its DNS is disabled (**LookupHostname**), a dash appears in this field.

Value in log	Description
type-specific	The type-specific information for the specified operation. The following are the returned values (the number and interpretation of type-specific values vary depending on the type of operation): BND The LDAP bind name enclosed in double quotes. A dash is used if the name cannot be resolved (normal case). UBD None ABN None ADD The ID of the certificate that was added to the PGP Keyserver. On signed requests, the ID of the certificate of the signer is also shown. A dash is used if there is no signer. Also included are the number of User IDs added, the number of signatures added, and the size of the added keyblock, in bytes. These are present even if the key already exists. MOD Logs the number of new userIDs and signatures (not including existing userIDs and signatures). SRC The number, in decimal, for the matches or hits returned by a search operation, followed by the sum of the sizes of all returned keyblocks, in bytes. Special case: SRC 0 key x bandwidth = Search for server information. Note that "0 0" is a normal search that returned no keys. SRP 1 key x bandwidth = size of prompts plus the size of the hash repetitions. SRD 1 key x bandwidth = size of the reconstruction data (or blob) DLT The ID of all deleted certificates, shown in double quotes, and, if the request requires a signature, the ID for the certificate of the signer. A dash indicates that the value is not applicable. DIS The ID of all disabled certificates, shown in double quotes, and, if the request requires a signature, the ID for the certificate of the signer. A dash indicates that the value is not applicable.

If double quotes or back-slashes exist in a value that is always enclosed in double quotes, they are escaped with back-slashes.

UNIX

The standard error (stderr) contains useful information about how the PGP Keyserver is performing. To generate this information, you must use the **-c** (configuration) or **-d** (debug) command line switch when you start the PGP Keyserver. For instructions, see "Starting the PGP Keyserver from the command line" on page 47.

For a more in-depth chronological view of all the requests the PGP Keyserver has processed, examine the Access Log file.

Sample Access Log File Entries (AccessLogFormat Version1)

The following are typical entries that appear in the Access Log File:

SRC 0 1998-05-26 23:46:21 0 171.169.27.75 xyz.nnn.com 1 -

SRC 0 1998-05-26 23:46:21 0 171.169.27.75 xyz.nnn.com 3 -

SRC 7 1998-05-26 01:16:03 0 171.169.27.75 xyz.nnn.com 1 -

ADD 7 1998-05-26 01:16:03 0 171.169.27.75 xyz.nnn.com 0x27535B7C40C97D40 -

SRC 0 1998-05-27 01:31:24 0 171.169.17.15 abc.nnn.com 1 -

SRC 0 1998-05-27 01:31:24 0 171.169.17.15 abc.nnn.com 13 -

Controlling the contents of the Access Log

Access Log File Cycling

Periodically, the PGP Keyserver copies the contents of the Access Log file to a new file, removes the contents of the Access Log file, and begins to record new access information in the empty Access Log file. This action, called cycling, prevents the Access Log file from becoming too large, and allows administrators to process the logs without interfering with PGP Keyserver operations.

The Access Log file settings in the configuration file control how frequently the PGP Keyserver cycles the entries in the Access Log file, and how many cycled files are retained on the PGP Keyserver. For more information, see "General configuration settings" on page 79.

Naming convention for cycled files

The cycled files are named by inserting the date, in the format YYYYMMDD, between the filename and extension of the Access Log File:

<filename>.<YYYYMMDD>.<extension>

For example, if the name of the Access Log file is cert.log, a cycled file created on April 30, 2000, would be named cert.20000430.log. If the Access Log file name does not have an extension, the date becomes the extension. For example, cert.20000430.

Cycled files are kept in the same directory as the Access Log file. If the PGP Keyserver attempts to cycle data and a file with today's date already exists, the PGP Keyserver assumes that the log files have already cycled, and no additional cycling occurs.

Retention period for cycled files

The maximum number of cycled files retained by the PGP Keyserver is controlled by the **CycleLogKeep** configuration setting. Each time the PGP Keyserver cycles, the PGP Keyserver counts the cycled files in the **AccessLogFile** directory and compares the total number of cycled files to the value for **CycleLogKeep**. When the number of cycled files exceeds the value for **CycleLogKeep**, the PGP Keyserver deletes enough old cycled files to satisfy the limit set by **CycleLogKeep**.

Note that the PGP Keyserver counts only those files that use the naming scheme for the current **AccessLogFile**, and that the date in the file name identifies the age of the file.

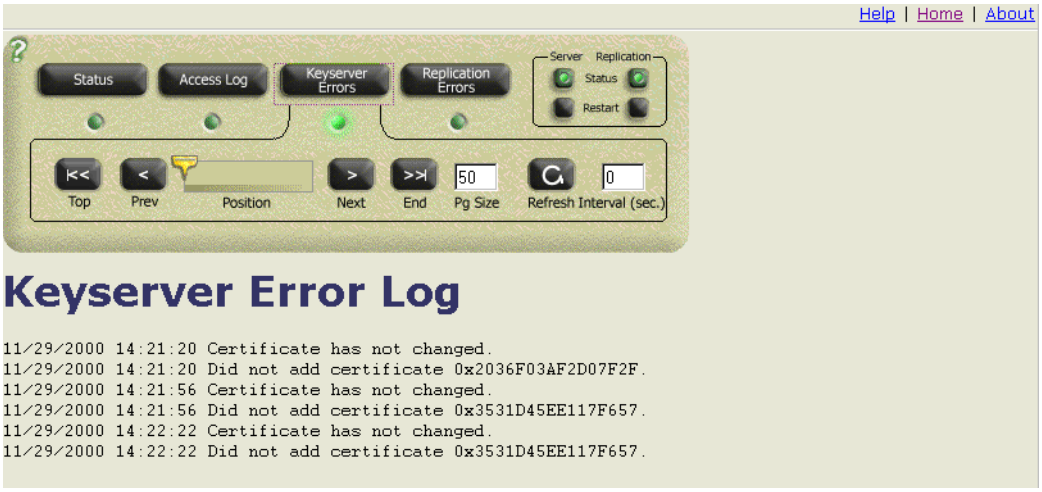
For details on the **CycleLogKeep** configuration setting, see "Logs to Keep - CycleLogKeep <number>" on page 79.

View the Keyserver Error log

To view the Keyserver Error log, click **Server Control** on the Web Console's left panel, and click **Keyserver Errors**. The PGP Keyserver displays the error log below the controls.

The Keyserver Error Log displays events that have occurred on the PGP Keyserver since the PGP Keyserver was last started or restarted. Entries are ordered by their date and time.

For a more detailed look at the errors that occur, consult the NT Event Log or the UNIX syslog. These logs will contain historical messages from before the last restart of the PGP Keyserver, which are not displayed in the Web Console's Error Log. The level of information is controlled by the PGP Keyserver's **LogLevel** configuration setting.



To learn more about the NT Event Log, see "Examining the NT Event Log: Windows NT" on page 69.

To learn more about the UNIX syslog, see "Examining the System Log File: UNIX" on page 70.

To learn more about the PGP Keyserver's LogLevel configuration setting, see "Logging Level for Event Log - LogLevel <level>" on page 80.

View the Replication Error log

To view the Replication Error log, click **Server Control** on the Web Console's left panel, and click **Replication Errors**. The PGP Keyserver displays the error log below the controls.

The Replication Error Log displays events that have occurred on the Replication Engine since the Engine was last started or restarted. Entries are ordered by their date and time.

For a more detailed look at the errors that occur, consult the NT Event Log or the UNIX syslog. These logs will contain historical messages from before the last restart of the PGP Keyserver, which are not displayed in the Web Console's Error Log. The level of information is controlled by the PGP Keyserver's **LogLevel** configuration setting.

NT Event Log and system log file

The level of information that appears in the NT Event Log and in the standard system log file, syslog, is controlled by the "**LogLevel** configuration setting.

Examining the NT Event Log: Windows NT

All entries in the NT Event Log that are associated with the PGP Keyserver and Engine have a source of PGP Keyserver. This distinguishes these messages from those generated by other processes.

Examining the System Log File: UNIX

The name and location of the system log file as well as the type of information it records is based on your system log configuration. For example, if you placed the line user.err /var/adm/user.log in the /etc/syslog.conf file, the PGP Keyserver details for the user data are logged to this file. For more detailed information on system log configuration, consult the UNIX man pages.

All entries in the system log that are associated with the PGP Keyserver and Engine have a source of **PGP Keyserver**. This distinguishes these messages from those generated by other processes. For a more detailed explanation of the cause and remedy of these errors, consult the next section.

Changing a PGP Keyserver or Engine configuration setting

The PGP Keyserver and Engine have many user-configurable settings that control how the PGP Keyserver and Engine work. These settings appear in the configuration file (see "The name and location of the configuration file" on page 72), and on the Web Console's left panel grouped in the following categories — General, Database, Secure Mode, Access, Policy, and Replication:

- The *general* configuration settings control log behavior, query limitations, and the Keyserver's primary port number.
- The *database* configuration settings control attributes of the Keyserver's database, including performance tuning options.
- The *secure mode* configuration settings control options associated with Secure Mode (LDAPS configuration).
- The *access* configuration settings control configuration values associated with access to the Keyserver, including access logs and client administrative passwords.
- The *policy* configuration settings define the policy requirements for your site, including the signatures that must be on a key before the PGP Keyserver will accept the key, and the signatures that are allowed to remain on a submitted key.
- The *replication* configuration settings configure the Keyserver and Replication Engine to control replication options, including file locations and LDAPS authentication.

Each setting is described in detail in Chapter 6, Keyserver and Engine Configuration Settings

The Web Console displays a descriptive name of each setting (for example, **Day to Cycle Log**) — the configuration file displays the actual name of the setting (for example, **CycleLogDay**). It isn't necessary to remember the exact name of the setting in the configuration file unless you elect to edit the configuration file manually. If you do want to know, click the book icon next to the setting on the Web Console — the help

for each setting displays both the descriptive name that appears on the Web Console and the name that appears in the configuration file. In addition, Chapter 6 includes a list of settings by name.

There are two ways to change a configuration setting:

- Edit the configuration settings via the Web Console.
- Edit the configuration file using your favorite text editor (advanced users).

You can change the settings or file at any time. The changes take effect after saving them and restarting the PGP Keyserver through the Web Console's **Server Control** panel.

Before you change any of the PGP Keyserver or Engine settings, please read the following notes:

- Before you make any changes to the settings on the Web Console's configuration panels (**General**, **Database**, **Secure Mode**, **Access**, **Policy**, and **Replication**), make sure that the configuration file displayed on the **Select Configuration file** panel is correct.
- When you change the value for an existing configuration setting, be sure to click the **Save Changes** button to save them and verify your changes. The configuration setting is updated in the setting's original location in the configuration file. It is also displayed on the Web Console.
- When you add a configuration setting that did not previously exist, the new setting and value appear at the bottom of the configuration file. It is also displayed on the Web Console.
- The PGP Keyserver's command line parameters override the values in the configuration file. For more information, see "Overriding settings in the configuration file" on page 72.

To change a PGP Keyserver or Engine parameter:

1. Display the appropriate Server Configuration panel on the Web Console's left panel (Select Configuration File, General, Database, Secure Mode, Access, Policy, or Replication).
2. Locate the configuration setting and make the appropriate change. For the Configuration File, click **Fetch**. For all other parameters, click **Save Changes**. The Web Console updates the configuration file (Complete). If it encounters errors, it displays an error message.

The name and location of the configuration file

All of the user-configurable settings are stored in a configuration file, pgpcertd.cfg (in UNIX, it is pgpcertd.conf). The file is normally stored in the following default location:

Windows:

C:\Program Files\PGP Corporation\PGP Keyserver\etc\pgpcertd.cfg

UNIX:

/opt/PGPkeysrv/etc/pgpcertd.conf

Note: If your configuration file is corrupted or deleted, you can use the master configuration file, pgpcertd-Master.cfg (in UNIX, pgpcertd-master.conf), to restore the original settings.

The full path name of the file is available and editable from the **Select Configuration File** option on the Web Console's left panel.

Editing the configuration file manually

If you edit the configuration file manually, note the following:

- Configuration setting keywords are not case-sensitive.
- Comments can be included by preceding a line with the pound sign (#).
- Blank lines are ignored.
- If a filename that appears in the configuration file contains one or more space characters, the filename must be enclosed in double quotes ("").

Verifying manual edits to configuration settings

If you edit a configuration file manually, use the following command on the command line to verify the validity of the file:

pgpcertd -g -c -f <filename>

Overriding settings in the configuration file

The PGP Keyserver's command line parameters override the values in the configuration file. To learn about the PGP Keyserver's command line parameters, see "Starting the PGP Keyserver from the command line" on page 47.

Using Secure Mode

The PGP Keyserver includes a Secure Mode that you can use to perform deletions and other administrative tasks. You can use Secure Mode for all interactions with the PGP Keyserver. For example, you can use Secure Mode to perform searches privately. When the PGP Keyserver is configured with Secure Mode required, the PGP Keyserver will not start unless it can successfully provide secure access by Transport Layer Security (TLS). TLS is a protocol based on SSL that provides encrypted and authenticated communications.

Note that the configuration wizard creates the key that Secure Mode uses and modifies the configuration file and keyring file accordingly.

To use Secure Mode:

1. On the Web-console PGP Keyserver's main page, click **Secure Mode** under **Server Configuration**, left panel.
2. Select **Required** under Secure Mode and click **Save Changes**.
3. Select **Server Control** under **Console** on the Web-console's main page.
4. Click **Restart**.
5. The PGP Keyserver displays the **PGP Enter Passphrase for Selected Key** window if the key has a passphrase.
6. Enter the passphrase of the PGP Keyserver's signing key and click **OK**.

   The PGP Keyserver is now in Secure Mode.

6 Keyserver and Engine Configuration Settings

This chapter includes a description of each of the PGP Keyserver's and Engine's user-configurable settings (for example, the **DBCacheSize** configuration setting identifies the size, in bytes, of the in-memory cache associated with the PGP Keyserver's database).

For instructions on how to change a configuration setting, see "Changing a PGP Keyserver or Engine configuration setting" on page 70.

Note: When you change the value for an existing configuration setting, the configuration setting is updated in the setting's original location in the configuration file. When you add a configuration setting that did not previously exist, the new setting and value appears at the bottom of the configuration file. New and changed values are also displayed on the Web Console.

For a list of configuration settings by group, see "Configuration settings, by group" on page 75.

For a list of configuration settings in alphabetical order, see "Configuration settings, alphabetized" on page 76.

Configuration settings, by group

The settings in the configuration file are grouped in the following categories:

- General configuration settings (page 79)
- Database configuration settings (page 81)
- Secure Mode configuration settings (page 83)
- Access configuration settings (page 86)
- Policy configuration settings (page 89)
- Replication configuration settings (page 92)

Configuration settings, alphabetized

The following table includes a brief description of each configuration setting. More complete information for a setting is located on the page noted in the right column.

AcceptReconstructionData Accept Reconstruction Data	Controls if the PGP Keyserver accepts key reconstruction data.	Policy	page 89
AccessLogFile Access Log File	Identifies the file where access statistics are logged.	Access	page 87
AccessLogDetails Items to Log in Access Log	Controls the level of statistics recorded in the Access Log File.	Access	page 88
AccessLogFormat Access Log Format	Identifies the access log file format.	General	page 81
AdminPassword Administrative Password	Identifies the password for the **AdminUsername** setting.	Access	page 88
AdminUsername Administrative User Name	Identifies the login name of the PGP client administrator.	Access	page 88
Allow Allow access by	Defines level of access for users.	Access	page 86
AllowSigID Allowed Certificate Signatures	Identifies signatures that are allowed when **TrimSigs** is turned on.	Policy	page 90
CacheEntries Cache Entries	Identifies the number of the database entries cached by the PGP Keyserver.	Database	page 83
CycleLogDay Day to Cycle Log	Controls when the Access Log File is cycled (archived).	General	page 79
CycleLogTime Time to Cycle Log	Controls the time of day that cycling of the Access Log File occurs.	General	page 79
CycleLogKeep Logs to Keep	Controls the number of old Access Log Files that the PGP Keyserver retains.	General	page 79
DBCacheSize Database Cache Size	Controls the database cache size in bytes.	Database	page 83
DefaultAccess Default Access	Defines default access.	Access	page 80
Directory Directory	Identifies where the database files are located.	Database	page 81
ForceSyncOnWrite Sync on Every Write	Controls if the PGP Keyserver will flush database changes to disk on every change.	Database	

IdleSyncTimeout Idle Sync Timeout	Directs the PGP Keyserver to save the database cache to disk after the PGP Keyserver has remained idle for a specified number of seconds.	Database	page 83
IndexMethod User ID Indexing	Identifies the method the PGP Keyserver uses to index User IDs on keys.	Database	page 82
LogLevel Logging Level for Event Log	Controls the level of event logging information sent to the operating system's logging mechanism.	General	page 80
LookupHostname Log client hostname	Directs the PGP Keyserver to perform a DNS lookup for each connection. If there is an Allow Host line, the Keyserver always performs a DNS lookup, regardless of this setting.	General	page 80
Mode Database File Permissions	Identifies the file permissions associated with the database.	Database	page 81
MustSigID Required Certificate Signatures	Identifies the signatures a key must have to pass the policy requirement.	Policy	page 89
PolicyFailures Action on Key Policy Failure	Controls if rejected keys are sent to the pending bucket or returns an error.	Policy	page 91
Port (Port)	Identifies the port to listen to for regular LDAP connections.	General	page 79
PublicKeyRing Public Keyring	Identifies the file that contains any private keys used by the Keyserver or Replication Engine, such as for **ReplicationSecureKeyID** or **SecureKeyID**.	Secure Mode	page 84
PrivateKeyRing Private Keyring	Identifies the PGP private keyring file that contains any private keys used by the Keyserver or Engine.	Secure Mode	page 85
RandSeedFile Random Seed File	Name of file to use to store persistent pseudo random seed.	Secure Mode	page 85
ReadOnly Database Access Mode	Controls read/write access to database entries.	Database	page 82
Replica Hosts to Replicate Database to	Identifies the location where the database contents are to be replicated.	Replication	page 92

ReplicationSecureKeyID Replication Secure KeyID	Identifies the key ID of the keypair to use as the key to authenticate the client side of all LDAPS connections and Web Console side of the Engine's connection.	Replication	page 93
RepLogFile Replication Log File	Identifies the log file where changes are recorded for replication.	Replication	page 92
SecureMode Secure Mode	Controls if Secure Mode is required, optional, or disabled.	Secure Mode	page 83
SecurePort Secure Port	Identifies the port to listen to for TLS connections.	Secure Mode	page 84
ServerSecureKeyID Server Secure KeyID	Identifies the key ID of the keypair to use as the PGP Keyserver's LDAPS key.	Secure Mode	page 84
SizeLimit Size Limit	Identifies the maximum number of matches returned for a query.	General	page 80
TempPath Temporary File Location	Identifies the fully qualified path to the directory that holds temporary files.	Replication	page 93
TimeLimit Time Limit	Identifies the maximum number of seconds allocated for a query.	General	page 80
TrimPhotoIDs Remove PhotoIDs	Instructs the PGP Keyserver to remove PhotoIDs from submitted keys.	Policy	page 89
TrimSigs Remove Unallowed Signatures	Instructs the PGP Keyserver to remove unauthorized signatures from submitted keys.	Policy	page 89
TrimUsers Remove Unallowed User IDs	Instructs the PGP Keyserver to remove unsigned user IDs from submitted keys.	Policy	page 89
User Run Server as Username	UNIX only. Identifies the user that the PGP Keyserver will run as, following initial startup.	Database	page 82

General configuration settings

This section describes the general configuration settings. They control log behavior, query limitations, and the Keyserver's primary port number.

Port - Port <*Port*>

Where <*Port*> is the port to listen to for regular LDAP connections. Valid values are from 1 to 65534. This defaults to port 389, the well-known port for LDAP. The port numbers for the **Port** and **SecurePort** configuration settings must be different, and no other program can use either of those ports. Use a port of -1 (negative one) to disable LDAP access to the Keyserver. If this is set, LDAPS must be properly configured or the Keyserver will not start.

Log Cycling

Day to Cycle Log - CycleLogDay <*frequency*>

This setting controls when and if the Access Log File is cycled (archived). Defaults to **Never**.

- To cycle the Access Log File weekly, select the day you want cycling to occur: **Monday**, **Tuesday**, **Wednesday**, **Thursday**, **Friday**, **Saturday**, or **Sunday**.
- To cycle the Access Log File every day of the week, select **Daily**.
- To disable cycling, select **Never** (the Access Log File continues to grow in size).

Time to Cycle Log - CycleLogTime <*time*>

This setting, which controls the time of day that cycling of the Access Log File occurs, uses a 24 hour clock (military time).

<*time*> is in the following format: HH:MM. HH is between 00 and 23, and MM is between 00 and 59. Defaults to 23:59.

Logs to Keep - CycleLogKeep <*number*>

Use this setting to control the number of old Access Log Files that the PGP Keyserver retains. When the number of old logs in the Access Log File directory exceeds the value for <*number*>, the PGP Keyserver deletes the required number of log files until the number of log files matches the value for <*number*>.

The value for <*number*> can be between 0 to 99. If you enter 0, the Access Log File is truncated when **CycleLogDay** and **CycleLogTime** occurs, and the data is not archived. Defaults to 10.

Server Query Limitations

Size Limit - SizeLimit <*size*>

Identifies the maximum number of matches to return for a given search operation. The default is 500 entries.

Time Limit - TimeLimit <*seconds*>

Identifies the maximum number of seconds, in real time, that the PGP Keyserver will spend processing a client search request. If the request is not fulfilled in the allotted time, a message is sent to the client indicating that the request has timed out. The default value is 300 seconds (5 minutes).

Logging Level for Event Log - LogLevel <*level*>

Identifies the degree of information recorded in the system log file, syslog (UNIX), or Event Log File (Windows NT). Note that this is not the same as the Access Log File. You can view the contents of the system log file or Event Log File to find out how the PGP Keyserver is performing. There are four levels of access, and they are hierarchically accumulative (that is, each level of logging details automatically includes all of the details provided by the lesser levels).

The following table describes valid values for the LogLevel setting:

error	Logs all error messages.
warning	Logs all errors and warning messages.
info	Logs all errors, warnings, and informational messages.
verbose	Logs all messages, including LDAP specific information.

Since the logging is output to the system log file or Event Log File, each entry generated by the PGP Keyserver has a source of "PGP Keyserver." This distinguishes these messages from those generated by other processes.

Log Client Hostname - LookupHostname yes | no

Directs the PGP Keyserver to perform a DNS lookup for each connection so that it can log both the host's IP address and its fully qualified name (for example, 123.abc.company.com).

If the **Allow** <*hostname*> configuration setting is used when **LookupHostname** is off, **LookupHostname** is turned on and a warning message is logged.

Default setting is **Yes**.

Access Log File Format - AccessLogFormat Version1 | Standard

Identifies the format to use for the access log. The default is Standard. Version1 is compatible with versions prior to PGP Keyserver 7.0. Note that Standard includes statistics not available in Version1 format (for example, Request Time and Request Bandwidth) that are useful for performing additional usage analysis.

Database Configuration Settings

The *database* configuration settings control attributes of the Keyserver's database, including performance tuning options.

Directory - Directory <*path*>

Identifies the relative or fully qualified path to the directory where the database files and associated index entries are stored. There is no default value. If a filename includes blank spaces, the name must be enclosed in quotes.

In Windows, defaults to "..\data".

Caution: In a UNIX environment, there must be a value for this setting, or the Keyserver will not start.

Database File Permissions - Mode <*file permissions*>

Identifies the file permissions associated with newly created database files.

Under Windows, this setting is always 0600 on Keyservers. The default setting, 0600, gives read and write access by the file's owner.

Under UNIX, this value must be expressed in octal, preceded with a zero. The default file permissions are 0600 (gives read and write access by the file's owner).

To set the permissions to be -rw-r--r--, as in chmod 0644, use Mode 0644.

To set the permissions to be -rwxrw-rw-, as in chmod 766, use Mode 0766.

Note: The umask value of the shell that pgpcertd runs affects permissions and can cause the permissions to be something other than what is specified in the Mode parameter.

Database Access Mode - ReadOnly yes | no

Controls if clients can read and write entries to the database, or if they are restricted to read-only access.

Read-only. When read-only mode is turned on, any attempt by a client to write to the database results in an "unwilling to perform" error message.

Read and write. By default, the read-only setting is turned off, which means clients have read and write access to the Keyserver.

This setting is useful when replicating data to multiple Keyservers, and you want to grant the ability to search for and retrieve entries, but prevent users from adding or modifying entries.

User ID Indexing - IndexMethod <*Method*>

Identifies the method used by the Keyserver to index User ID's on keys. The default is 'Substring'.

The method is either "Word" or "Substring." When User ID's are indexed, the Keyserver will either index only the word components (including the separate parts of the email address and real name) or it will index all the substrings in the User ID. If the **IndexMethod** is changed, the database must be recreated; if it is not, the Keyserver issues a warning message that it is ignoring the new **IndexMethod**.

To change IndexMethod, follow the steps below:

1. Shut down the PGP Keyserver service in the Windows Services Control Panel.
2. Export the existing database (**Start—>Programs—>PGP Keyserver—>PGPexport**)
3. Change the **IndexMethod** configuration setting on the Web Console's Database panel.
4. Recreate the database. To do so, remove the data files (all of the files in the Keyserver's data directory) or change the **Directory** configuration setting, then restart the Keyserver.
5. Use PGPimport to add the keys back to the Keyserver (**Start->Programs->PGP Keyserver->PGPimport**).

User - User <*Username*>

UNIX only. Identifies the user that the Keyserver will run as, following its initial startup. This should be set to the name of a UNIX account. The default is pgpkserv.

The Keyserver will drop root permissions after performing any privileged operations required to start, such as binding to port numbers less than 1024.

Sync on Every Write - ForceSyncOnWrite yes | no

Controls if the Keyserver flushes database changes to disk on every change. Valid settings are **yes** and **no** - defaults to **yes**. **No** provides better performance, but presents the possible risk of losing some changes if power is lost before changes are flushed to disk.

Cache Settings

Cache Entries - CacheEntries <*number of entries*>

Identifies the number of entries (that is, keys and their associated user IDs) that are cached by the Keyserver. The default cache size is 50 entries.

DB Cache Size - DBCacheSize <*size*>

Identifies the size, in bytes, of the in-memory cache associated with the database. Increasing the database cache size uses up additional memory but can dramatically improve performance, especially when modifying database entries. The default size is 1,000,000.

Idle Sync Timeout - IdleSyncTimeout <*seconds*>

Identifies the number of seconds the Keyserver can remain idle before any new entries in the database cache are saved to disk. After the time-out expires, the contents of the current cache are examined to see if any new entries have been added, and then this information is saved to disk. The default is 10 seconds.

Secure mode configuration settings

Use the following configuration settings to operate the Keyserver in *Secure Mode*. Use Secure Mode to perform administrative functions such as the deletion of keys. When the Keyserver is in Secure Mode (that is, the value for the **SecureMode** setting in the configuration file is **Required**), the PGP Keyserver cannot start unless it can successfully provide secure access by *Transport Layer Security (TLS)*. TLS is a protocol based on SSL that provides encrypted and authenticated communications.

Secure Mode - SecureMode <*Mode*>

Where <*Mode*> is **Disabled**, **Required**, or **Optional**. Default is Disabled.

- **Disabled**. Turns off Secure Mode.

- **Required**. The Keyserver cannot start unless it can successfully provide secure access (Transport Layer Security (TLS)). When this setting is used, there must be a Keyserver key in the **PublicKeyRing** and **PrivateKeyRing** keyring files. In addition, the operator must enter the passphrase for the secret Keyserver key each time the Keyserver starts or the Keyserver key must be passphraseless.
- **Optional**. If the Keyserver is started in auto-start mode, then the Keyserver enables TLS if TLS can be enabled without user intervention (that is, no passphrase is required). Otherwise, the Keyserver starts with TLS disabled. If the Keyserver is not started in auto-start mode, then this is the same as setting **SecureMode** to **Required**; a passphrase dialog will be presented, if necessary.

  Auto-start mode is used when the -s command line argument is present or when the Keyserver is started as an NT service.

  TLS can only be started without user intervention if the **ServerSecureKeyID** is set to a valid key and the key does not have a passphrase.

Note that under certain conditions the Engine turns this setting off (**Disabled**).

Note: When you change the **SecureMode** configuration setting to **Required**, you must restart the Keyserver; sending the HUP signal to the Keyserver or using the **Restart** feature in the Web Console's **Server Control** panel will not enable the setting.

Secure Port - SecurePort <*Port*>

Where <*Port*> is the port to listen to for TLS connections. Valid values are from 1 to 65534. This setting defaults to port 636, the well-known port for LDAP over TLS (LDAPS). Use the -1 (one) option to disable the use of the port. However, the Keyserver will not start if Port and Secure Port are both set to -1.

Server Secure KeyID - ServerSecureKeyID <*KeyID*>

Identifies the key ID of the keypair to use as the Keyserver's LDAPS key. This key must be in the keyring files specified by the **PublicKeyRing** and **PrivateKeyRing** configuration values. If this is not specified, the Keyserver does not start.

<*KeyID*> is either a 32 or 64 bit PGP KeyID. The KeyID must have the prefix 0x which is followed by a hexidecimal value. For example, 0x9615A02DBBE1E0E2.

Use this parameter to identify the KeyID of the Keyserver's TLS key. If autostart mode is active (that is, you use the -s command line option or you start the Keyserver as an NT Service), the Keyserver disables Secure Mode unless this parameter is defined.

If this key does not have a passphrase, the Keyserver does not prompt for a passphrase and continues execution.

Windows

If the Keyserver is running as an NT service, Secure Mode is optional, and the key requires a passphrase, the Keyserver disables the secure port.

If the Keyserver is running as an NT service, Secure Mode is required, and the key requires a passphrase, and the NT service is not configured to interact with the desktop, the Keyserver exits. This reflects a misconfiguration in the NT service.

Under normal circumstances, when Secure Mode is required and the key has a passphrase, the Keyserver prompts for the passphrase.

Public Key Ring - PublicKeyRing <*filename*>

Where <*filename*> is the fully qualified pathname to a valid PGP public keyring file. If a relative pathname is used, it is relative to the directory from which the Keyserver was started.

The Keyserver looks in this keyring file for keys specified by the **KeyID** configuration setting. Any key used as the Keyserver's TLS key or specified by an '**Allow Keyid**, '**MustSigID**, or '**AllowSigID** configuration value can be located in this file.

When the Keyserver is in Secure Mode (that is, the value for the **SecureMode** setting in the configuration file is **Required**), the Keyserver cannot start unless it can successfully provide secure access by *Transport Layer Security (TLS)*. TLS is a protocol based on SSL that provides encrypted and authenticated communications.

Note that when the **SecureMode** setting in the configuration file is set to **Optional** but specific parameters are not defined (for example, **ServerSecureKeyID**), **SecureMode** and the **SecurePort** are disabled.

Defaults to "..\etc\pgpkeyserver-pubring.pkr".

Private Key Ring - PrivateKeyRing <*filename*>

Where <*filename*> is the fully qualified pathname to a valid PGP private keyring file. If a relative pathname is used, it is relative to the directory from which the Keyserver was started. Use PGP to create this file. To enable support for TLS, the private portion of the Keyserver's TLS key must be in this keyring.

Defaults to "..\etc\pgpkeyserver-secring.skr".

Random seed file - RandSeedFile <*filename*>

Name of file to use to store persistent pseudo random seed.

Windows

Defaults to Windows NT directory (for example, C:\WINNT\Profiles\All Users\Application Data\PGP Corporation\PGP\randseed.rnd).

UNIX

Defaults to ./randseed.bin.

Access configuration settings

Note: When configuring access controls, there are two important parts. First, operations must be permitted by hostname or IP address (**Allow IP**..., **Allow Host** ..., or **DefaultAccess** ...). Second, certain operations such as deletes must also be authenticated using a PGP key (**Allow KeyID** ...). Any operation which requires an **Allow KeyID** ... directive must also be allowed explicitly by **Allow IP** or **Allow Host** or implicitly with **DefaultAccess**.

The *access* configuration settings control configuration values associated with access to the Keyserver, including access logs and client administrative passwords.

Default Access - DefaultAccess none | search | read | add | delete | all

Identifies the default level of access granted to all users who are not covered by the **Allow IP** or **Allow Host** setting. The following table describes valid values for **DefaultAccess**:

none	Denies all access to default users.
read	Allows default users to query and retrieve keys from the Keyserver.
add	Allows default users to query and retrieve keys and to add new keys to the Keyserver.
delete	Allows default users to retrieve, add, and delete keys from the Keyserver. This access level does not permit generic LDAP deletes (only PGP Key deletes). Key deletion also requires an "Allow KeyID ____ delete" directive.
admin	Allows default users to perform all of the above functions. They can also update administrative settings such as PGP client Preferences, PGP Groups and PGP Key Reconstruction information. Key deletion and admin-restricted operations are enabled with a corresponding "**Allow KeyID** _____ admin" directive. **AdminUsername** and **AdminPassword** are also required for updating PGP client preferences.

Allow Access By - Allow

Grants access for specific actions on the Keyserver to users.

To change this setting, click **Access** on the Web Console's left panel, and click **Add** next to **Allow Access By**. Make the required changes, click **Add**, then click **Save Changes**. Use the up-arrow links in the **Allow Access By** to adjust the order of Allow lines.

Users are identified in the following manner:

- By choosing IP or host and selecting an access level as described in "Default Access - DefaultAccess none | search | read | add | delete | all" on page 86.

Note: To allow updates to preferences, you must set an Allow <*host*> admin or Allow <*IP*> admin, plus the person performing the update must use the correct **AdminUsername** and **AdminPassword** when performing the preference updates in PGPadmin.

- By choosing KeyID:

  You may choose Delete or Admin.

  Delete allows a request to disable or delete keys on the Keyserver if the connection is strongly authenticated with the specified KeyID. The special value "self" can be used in place of an actual KeyID to allow any key-owner to delete or disable their own keys.

  Admin allows an administrator running PGPKeys to upload and modify administrative settings to the Keyserver, including PGP Group. The administrator must be the owner of the key specified by the indicated KeyID.

To find out how to identify the 64-bit KeyID, see "Extracting key IDs: The PGP Key ID utility" on page 38.

Note: For the Allow KeyID line to be honored, the IP address, Hostname, or **DefaultAccess** must also allow the requested operation. Use **Allow IP**, **Allow Host**, or **DefaultAccess**.

The permission granted by the first "allow host" or "allow IP" line that is encountered takes precedence over all subsequent lines. This means that once you grant a certain type of permission to a user, any subsequent permissions that conflict with the initial level of permission are ignored. To avoid any conflicts, place the most specific items first. For example, you should define complete host names (admin.pgp.com) before partial host names (*.nnn.com or server??.nnn.com).

Access Log File - AccessLogFile <*filename*>

Identifies the relative path or absolute pathname for the Access Log File (by default, cert.log in the Keyserver's logs directory).

Items to Log in Access Log - AccessLogDetails *<type>*

Controls the type of Keyserver operations recorded in the Access Log File. If you are editing the configuration file manually, you must list each operation that you want recorded in the Access Log File, and you must separate operations with a space. The following table describes valid values for **AccessLogDetails**.

none	No information is recorded in the access log (Web Console automatically sets when nothing is checked).
bind	Records bind operations.
unbind	Records unbind operations.
abandon	Records all abandon operations. This setting is on by default.
add	Records all add operations. This setting is on by default.
modify	Records all modify operations. This setting is on by default.
search	Records all search operations. This setting is on by default.
delete	Records all delete operations. This setting is on by default.
ldap	Records all LDAP operations that are not normally handled by the Keyserver.
all	Record all of the operations listed above (Web Console automatically sets when all operations are checked).

Administrative Username - AdminUsername *<username>*

Identifies the login name of the PGP client administrator. Use in conjunction with the **AdminPassword** setting to identify the user who can modify Preferences for PGP clients who store this information on this Keyserver.

Administrative Password - AdminPassword *<password>*

Identifies the PGP client administrator's password—use in conjunction with the **AdminUser** setting. This password may be in clear text or in Netscape's SSHA format (salt-with-SHA-1):

http://developer.netscape.com/docs/technote/ldap/pass_sha.html

To improve security, use SSHA-1 format. SSHA-1 format is always used when the password is set in the Web Console. To change the password, the administrator can use the Web Console or the configuration wizard.

Policy configuration settings

The certificate policy configuration settings define the policy requirements for your site. Use these settings to identify which signatures must be on a key before the Keyserver will accept the key, and which signatures are allowed to remain on a submitted key.

Remove Unallowed Signatures - TrimSigs yes | no

Allows you to remove unauthorized signatures from the UserIDs on a key before it is stored on the Keyserver. When this setting is turned on (that is, set to **yes**), all signatures except the owner's and those listed by the **MustSigID** and **AllowSigID** settings are trimmed from the key. The default setting is **no**.

Note: Do not use this setting unless the **MustSigID** or **AllowSigID** setting is used.

Remove Unallowed User IDs - TrimUsers yes | no

Allows you to remove unauthorized user IDs from a key before it is stored on the Keyserver. When this setting is turned on (that is, set to **yes**), the **MustSigID**, **AllowSigID**, and **TrimSigs** settings are enforced first. Then user IDs that have only a self-signature remaining are trimmed from the key. The default setting is **no**.

Remove PhotoIDs - TrimPhotoIDs yes | no

Allows you to remove a PhotoID from a key before the key is stored on the Keyserver. When this setting is turned on (that is, set to **yes**), PhotoIDs, which can be quite large, are removed from keys. Use this setting to reduce the size of the data stored by the Keyserver. The default setting is **no**.

Accept Reconstruction Data - AcceptReconstructionData yes|no

Controls if the Keyserver will accept key reconstruction data uploaded by PGP clients. Valid settings are **yes** and **no**. If a client attempts to post reconstruction data when this setting is set to **no**, the client will receive a "Permission Denied" error. The default setting is **on**.

Required Key Signatures - MustSigID <*keyID*>

Identifies the 32 or 64-bit key IDs for required signatures on a client key.

If you are using the Web Console to add signatures, click **Policy**, scroll down to **Required Key Signatures**, and click **Add**. Enter a keyID in the box, and click **Add**. To add additional required signatures, repeat this process.

If you want to require multiple signatures, list each of the required signatures on a single line. To require at least one of two or more signatures on a key, list each of the optional keys on a separate line. For example:

MustSigID 0x1234567812345678 0x12345678

In this case, the key must be signed by both keys before it is accepted by the Keyserver. Let's look at another example:

MustSigID 0xabcdef0123456789

MustSigID 0xfedcba987654321

In this case, the key must be signed by at least one of the keys in order to pass the policy requirement.

To find out how to identify the 64-bit KeyID, see "Extracting key IDs: The PGP Key ID utility" on page 38.

Note: Before you start the Keyserver, make sure all of the required certificates are stored on the Keyserver or in the Keyserver's public keyring (See "Public Key Ring - PublicKeyRing <filename>" on page 85).

Allowed Key Signatures - AllowSigID <*keyID*>

Lists the 32 or 64-bit key IDs for signatures that are considered allowable when the **TrimSigs** setting is turned on. When trimming signatures, only the owner's signature and those listed by the **MustSigID** and **AllowSigID** settings are allowed to remain on the key. All other signatures are trimmed from the key before it is placed on the Keyserver.

To add allowed signatures via the Web Console, click **Policy** on the Web Console's left panel, scroll down to **Allowed Key Signatures**, click **Add**, enter the keyID, and click **Add**.

You can add multiple **AllowSigID** lines; each line is treated with equal significance.

To find out how to identify the 64-bit KeyID, see "Extracting key IDs: The PGP Key ID utility" on page 38.

Note: Before you start the Keyserver, make sure all of the required certificates are stored on the Keyserver or in the Keyserver's public keyring (see "Public Key Ring - PublicKeyRing <filename>" on page 85). When the configuration file is validated, it cannot check its database; PGP Corporation recommends updating the Keyserver's keyring before changing the configuration.

Action on Key Policy Failure - PolicyFailures pending | error

Allows you to specify if keys rejected due to policy failure are sent to the pending bucket for further evaluation, or if they are tossed with an accompanying error message. If set to **Send to Pending Bucket**, the key is stored in the pending bucket. If set to **Return Error Message**, the key is ignored and an error message is generated. The **Return Error Message** setting is useful for sites that do not want to maintain a pending bucket. The default setting is **Send to Pending Bucket**. In the configuration file, the settings are **pending** and **error**.

You can search the pending bucket using the PGPkeys search dialog. See "Resolving keys in the pending bucket" on page 63.

Certificate policy configuration matrix

The following matrix is designed to help you understand the ramifications of using the policy configuration settings in combination with one another.

MustSigID	AllowSigID	TrimUserID	TrimSigs	Keyserver Results
Not set	Any or no value	No	No	The Keyserver accepts all keys regardless of how they are signed, and performs no trimming.
Set	Any or no value	No	No	The Keyserver accepts any certificate with at least one User ID signed with a key in the **MustSigID** list. No trimming is performed.
Set	Any or no value	No	Yes	The Keyserver accepts any certificate with at least one user ID signed with a key in the MustSigID list. All User IDs are accepted, but only the owner's signature and those in the **AllowSigID** and **MustSigID** lists are retained; all other signatures are removed from the key.
Set	Any or no value	Yes	Yes	The Keyserver accepts any certificate with at least one User ID signed with a key in the **MustSigID** list. Only User IDs signed by a key listed in the **MustSigID** or **AllowSigID** lists are accepted; all other user IDs are trimmed. Only the owner's signature and those in the **AllowSigID** and **MustSigID** lists are retained; all other signatures are removed from the key.

A key may be revoked if the key is compromised or old. If a key is revoked, and the key has a signature from a **MustSigID**, the key still passes policy and is allowed in the database. This is so that revoked signatures can propagate to clients that already have the key with the positive signature on it. You can disable the key if this behavior is not desired.

Replication configuration settings

To learn about the replication process and replication configurations, see Chapter 7, The Replication Process."

If you plan to support replication (database entries stored on a master Keyserver are mirrored on other slave Keyservers on the network), you must identify the other Keyservers that will operate as replicas. The Engine, PGPrepd, can then replicate the required data and transfer the data to the replication Keyservers.

Note: When you use the Engine, you identify the Keyservers that will hold the replicated database information (**Replica**), and the name of the replication log file (**RepLogFile**). If you do not specify both of these configuration settings, the replication will not work. Note that the Engine uses the same configuration file as the Keyserver in the default installation.

The following are the relevant configuration settings for replication.

Hosts to Replicate Database to - Replica [<*protocol*>://] <*hostname*> or <*IPaddress*> [:<*port*>]

Identifies the protocol, hostname or IP address, and an optional port number for the slave machine that must be updated whenever a change in the contents of the master database occurs. Valid protocols are LDAP, LDAPS, and HTTP. If the protocol HTTP is used, the replica Keyserver must be running an MIT style public keyserver. If you do not specify a port number, the default port for that protocol is used (389, 636, and 11371 respectively). The protocol defaults to LDAP.

To add entries via the Web Console, click **Replication** on the Web Console's left panel, click **Add**, enter the required information, click **Add**, then click **Save Changes**.

Replication Log File - RepLogFile <*filename*>

Gives the fully qualified path for the log file that records all changes to the database on the master Keyserver. The Engine program consults this file to identify the data that must be replicated to the slave Keyservers.

When you set up a replication scheme for your Keyserver, note that the replication sends any new changes that occur after the replication is implemented. If your master Keyserver contains existing entries, you must export the entries to the other Keyservers to ensure that they are in each Keyserver's database.

For details on how to export data from one Keyserver to another, see "Exporting keys from the PGP Keyserver: The PGP Export utility" on page 39."

Note: If a filename includes one or more spaces, you must enclose the entire name in quotes.

Temporary File Path - TempPath <*path*>

Specifies the fully qualified path to the directory used for storing temporary files. This path name must be quoted if it contains any spaces. This is managed for you if the Web Console is used to make changes.

If this parameter is not specified in the configuration file, the value of the environment variable **TMP** is used. If **TMP** does not exist, **TEMP** is used. If these environment variables do not exist, /tmp is used under UNIX and C:\ is used under Windows.

Replication Secure KeyID - ReplicationSecureKeyID <*KeyID*>

Identifies the key ID of the keypair used to authenticate a Replication Engine to a remote PGP Keyserver.

This key must be in the keyring files specified by the **PublicKeyRing** and **PrivateKeyRing** configuration values. If this is not specified, the first public/private keypair found in the keyring is used.

<KeyID> is either a 32 or 64 bit PGP KeyID. The KeyID must have the prefix 0x which is followed by a hexidecimal value. For example, 0x9615A02DBBE1E0E2. Use the pgpkeyid utility, available from the start menu, to identify the 64 bit PGP KeyID.

This key is particularly useful when delete operations are received over LDAPS, because such a delete operation does not contain a PGP signature (the authentication is done at the TLS layer rather than using any authentication included as part of the payload transmitted over LDAP). By configuring the remote Keyserver to allow such operations from this keyID, these operations can be propagated correctly over LDAPS.

7 The Replication Process

This chapter describes the replication process.

PGP Keyserver and Engine configurations

If your installation is large or your users are in a number of different locations, a single PGP Keyserver may not meet your users demand for keys. As a result, you may require a PGP Keyserver on a number of systems. When you install a PGP Keyserver on multiple systems, the Replication Engine synchronizes the databases. The Replication Engine runs on the same system as the PGP Keyserver, and sends new and modified keys to other Keyservers. You can run the Replication Engine on any of the systems where a PGP Keyserver resides.

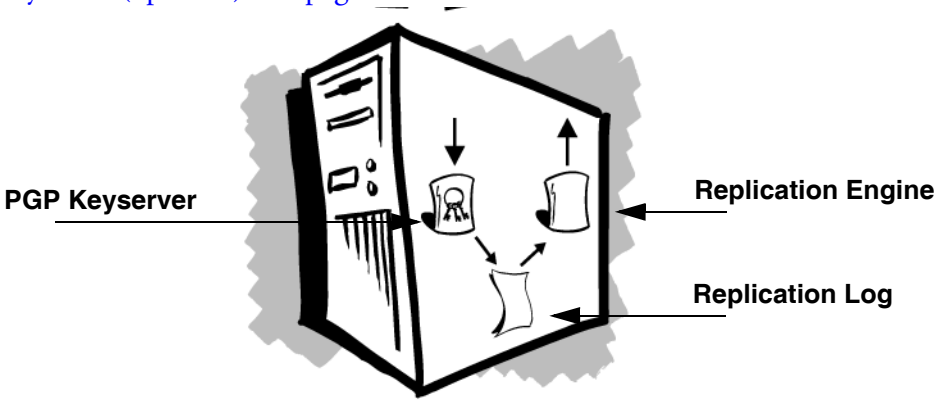
Keyservers are either Master Keyservers or Slave Keyservers.

- Master Keyservers perform all PGP Keyserver functions. A PGP Keyserver, replication log (a log where new and modified keys and destination Keyservers are recorded), and a Replication Engine reside on each master PGP Keyserver.
- Slave Keyservers perform search functions.

You can add slave or master PGP Keyservers to the same physical location or remote locations. Multiple master Keyservers (Keyservers that can perform all PGP Keyserver functions), are considered peers.

To learn more about the replication process, see "About replication" on page 99.

For step-by-step instructions to add a new PGP Keyserver, see "Adding an additional PGP Keyserver (optional)" on page 33.



Requests for deletions and disables must be strongly authenticated using LDAPS or a Signed Request over LDAP. In either case, the PGP Keyserver that is receiving the replication must be set up to allow deletions from the key that strongly authenticated

the deletion or disable request. In other words, if deletions or disables are to succeed, the master and slave PGP Keyservers must have the same Allow keyid Ox???? delete lines in their configuration files.

PGP Keyserver configurations

Before you install additional PGP Keyservers, consider the locations that the PGP Keyservers will service and the load in each location. The following section describes a few typical scenarios.

PGP Keyserver configuration models

The following PGP Keyserver configuration models represent only a few of the potential PGP Keyserver configurations. Each of the models identify the PGP Keyserver relationships (master, slaves, and peers), and the efficiency and tolerance rating for the models.

efficiency: A model with high efficiency has a minimum number of replications. As a result, it uses less network bandwidth, CPU load, and PGP Keyserver load. The fewer the replications, the higher the efficiency. Efficiency is high, medium, or low.

tolerance: A model with high tolerance has a large number of redundant replications. The more redundancy a model has built into it, the higher its tolerance. An example of a configuration with low tolerance is a single PGP Keyserver; if that PGP Keyserver goes down or has a problem, there is no backup. Tolerance is high, medium, or low.

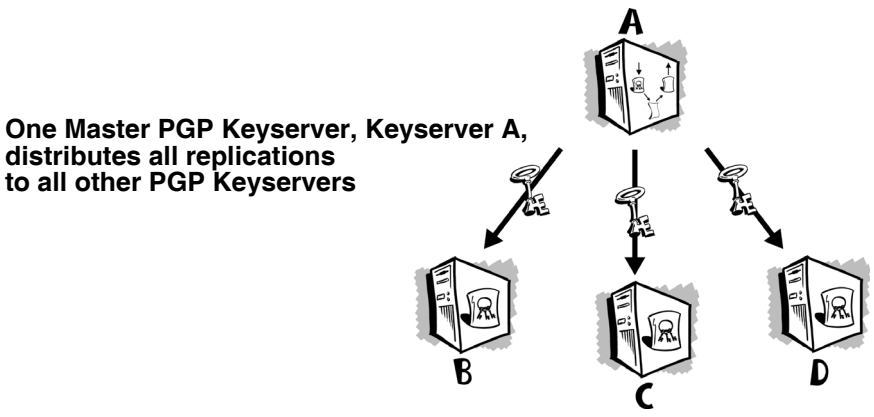
Note: When a PGP Keyserver replicates a modified key, the whole key is replicated, not just the changes.

Master slave model (high efficiency, medium tolerance)

Use for load balancing or distribution in a single organization. Can also use to ensure 100% availability. This model consists of one master PGP Keyserver (Keyserver A) with multiple slave or backup PGP Keyservers (Keyserver B, Keyserver C, Keyserver D, and so on). All PGP Keyservers are available for search requests, but only the master, Keyserver A, can perform adds, deletions, and disables.

When this model is used, all users can access Keyserver A, but they may not be able to access each of the slave PGP Keyservers (accessible PGP Keyservers appear in the PGPkeys Search Window). For example, each of the slave PGP Keyservers may service a specific division, and the users in that division may access PGP Keyserver A and their division's slave PGP Keyserver only. In another example, some users may know about two PGP Keyservers, and be told to use one as their primary PGP Keyserver, and the other as their backup PGP Keyserver.

Tolerance is medium because there are backup systems for queries but only one system for adds, deletions, and disables. Efficiency is medium because this configuration requires the minimum number of replications.



Star model (medium efficiency, medium tolerance)

Use for load balancing and 100% availability. This model consists of one master PGP Keyserver, Keyserver A, that distributes all replications to all other PGP Keyservers (Keyserver B, Keyserver C, Keyserver D, and so on). All of the PGP Keyservers can receive adds, deletions, and disables. However, there is no direct communication between Keyserver B, Keyserver C, and Keyserver D. These PGP Keyservers send adds, deletions, and disables to Keyserver A, and Keyserver A replicates these changes to the other PGP Keyservers.

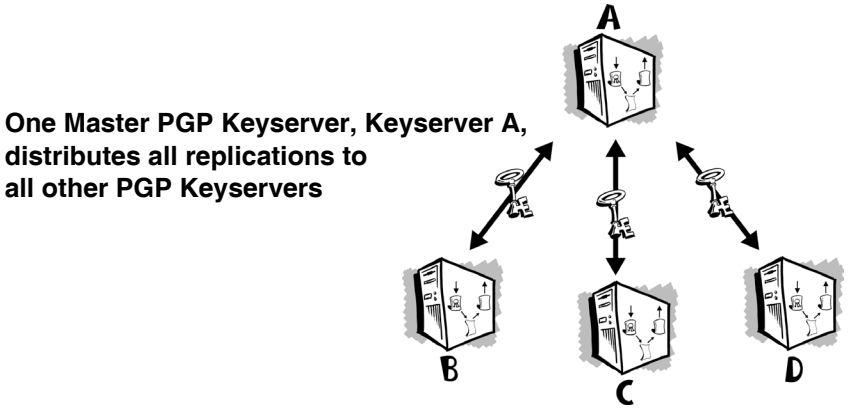
Efficiency is medium because there is two-way replication. (In this case if Keyserver B receives an add, Keyserver B sends the add to Keyserver A, Keyserver A sends the add back to Keyserver B, and also sends the add to Keyserver C and Keyserver D).

This model is easier to use than the Master-Slave model, because you can use any of the PGP Keyservers for any purpose.

Let's look at an example. Keyserver A is in New York, Keyserver B is in Australia, Keyserver C is in San Francisco, and Keyserver D is in Germany.

If you use the Master-Slave model, all adds, deletions, and disables must occur in New York. You can perform other PGP Keyserver functions locally, but you must perform your adds, deletions, and disables remotely.

If you use the Star model, you can perform all PGP Keyserver functions locally; the delay in PGP Keyserver synchronization is minimal (that is, limited to the time it takes to update changes via the Replication Engine).



We recommend that Keyserver replication be used in either master/slave or star replication models. Replication models that include a ring of three or more master Keysevers have the potential of allowing replication loops of certain uncommon types of key-update operations. In practical use, such a scenario can occur in rare circumstances that must include high network load and/or low availability, occurring simultaneously with these uncommon types of key-update operations.

Although such a replication loop causes no additional disk space usage for databases, it can create unnecessary network traffic and disk space consumption for access logs. If such a situation occurs, it can be easily remedied by stopping the Replication Engines, removing their temporary files (see "Temporary File Path - TempPath <path>" on page 93) and restarting the Replication Engines.

Use of the master/slave or star replication models detects and prevents such replication loops in progress and avoids the replication of problematic key-updates when they have been sent directly from a known replica's IP address.

Examples of different server configurations

International company with offices in the U.S. and Europe. A PGP Keyserver in the U.S. handles all PGP Keyserver requests that originate in the U.S., and a PGP Keyserver in Europe handles all PGP Keyserver requests that originate in Europe. The Replication Engine maintains the two databases. When a key is added to either of the PGP Keyservers, the Replication Engine copies the key to the other PGP Keyserver. The two PGP Keyservers are peers.

Domestic company with one large location, four divisions. One PGP Keyserver is installed in each division and handles all PGP Keyserver requests that originate in that division. The Replication Engine on each PGP Keyserver replicates new or modified keys to the master Keyserver, which may be a machine of its own, such as at company headquarters, or which may be one of these four Keyservers. It then replicates the

changes to all the replica Keyservers. This star model configuration is primarily used for load balancing, to allow each division to search keys independently using their own hardware and networking resources.

Domestic company with one large location. Users throughout the office submit keys to Keyserver A, and Keyserver A replicates all new or modified keys to Keyserver B, Keyserver C, and Keyserver D. This configuration allows the company to distribute the load of PGP Keyserver requests and maintain a low bandwidth replication model.

Small domestic company, one location. Keyserver A, the master PGP Keyserver, replicates to Keyserver B, the slave PGP Keyserver. If Keyserver A fails, users can automatically go to Keyserver B for their requests. This configuration might be used to ensure 100% PGP Keyserver availability (fault tolerance).

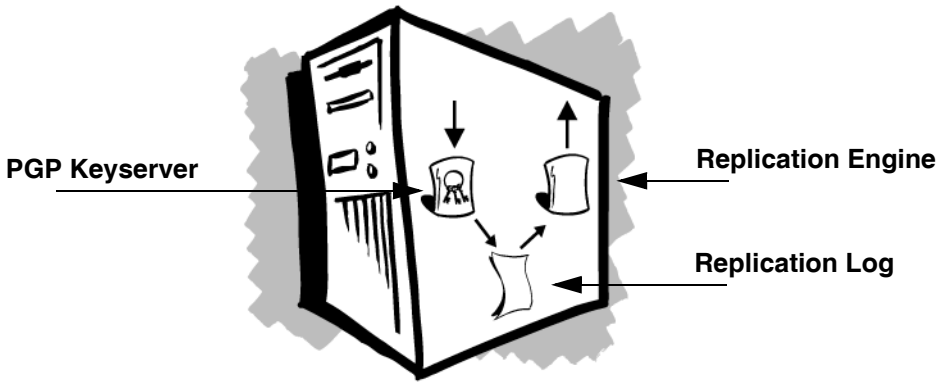
About replication

With the exception of PGP Keyservers B, C, and D in the Master-Slave model, all servers run a Replication Engine and replicate adds, deletions, and disables to the servers that appear in the replica line of their configuration file. Note that each Replication Engine must be started on each server, even in a master-slave relationship. If the server's Replication Engine is not started, it can receive adds from other servers, but it cannot forward adds to other servers.

Let's take a closer look at how replication occurs between three PGP Keyservers. Keyserver A is the master server, and Keyserver B and PGP Keyserver C are slave servers. The slave PGP Keyservers, Keyservers B and C, can perform all PGP Keyserver functions except adds, deletions, and disables. The master PGP Keyserver, Keyserver A, can perform all PGP Keyserver functions, including adds, deletions, and disables.

- When Keyserver A receives a new or modified key, Keyserver A checks the replica line in its configuration file to identify the PGP Keyservers that it replicates or sends new or modified keys to. The names of Keyserver B and Keyserver C appear in Keyserver A's configuration file.
- Keyserver A places copies of the new or modified key in a queue (that is, its replication log or *replog*) for Keyservers B and C.

- Keyserver A's Replication Engine sees that there are keys in the queue, and looks for the machines that it must send the keys to. Keyserver A finds the key designated for Keyserver B, and sends it to Keyserver B. Keyserver A finds the key designated for Keyserver C, and sends it to Keyserver C. Since Keyservers B and C are slave PGP Keyservers, they do not replicate the keys to other PGP Keyservers.



When does replication occur?

Replication occurs almost instantaneously, as long as the Replication Engine and target PGP Keyserver are running.

- If the Replication Engine is not running, the target PGP Keyserver is updated as soon as the Replication Engine is restarted.
- If the target PGP Keyserver is not running, the Replication Engine continuously looks for the machine. When the target PGP Keyserver becomes available, it sends the keys that are queued for that PGP Keyserver.

Note that if the target PGP Keyserver and Replication Engine are both down, the replication information is not lost.

How and where is the replication log (replog) maintained?

If there is a master PGP Keyserver, the master PGP Keyserver maintains the replication log (replog). The master PGP Keyserver receives new or modified keys, writes them to its own database, places the keys in the replog, and sends the keys to the slave PGP Keyservers. The slave PGP Keyservers do not maintain a replog, but they do have a database.

The Replication Engine moves entries from the master PGP Keyserver's replog into a temporary replog file for each remote slave PGP Keyserver, then sends the file to the slave PGP Keyservers. The **TempPath** configuration parameter controls the location of the temporary files.

In peering or star models, each PGP Keyserver maintains a replog and database; the database is updated by the replication process on each machine.

Can two PGP Keyservers replicate to each other?

What happens if Keyserver A replicates to Keyserver B, and Keyserver B replicates to Keyserver A? Does the Replication Engine get caught in an infinite loop?

When a new or modified key is added to Keyserver A, Keyserver A adds the key to the database and replog. The Replication Engine sees the key in the replog, knows that it replicates keys to Keyserver B, and sends the key to Keyserver B. Keyserver B adds the key to its database and replog.

Keyserver B's Replication Engine sees the new key in the replog, knows that it replicates keys to Keyserver A, and sends the key to Keyserver A. When Keyserver A receives the key, it sees that it is not a new key, and it does not write the key to the database or replog.

Another scenario might be that Keyserver B has a different version of the key in its database. Keyserver A sends a new key to Keyserver B, and Keyserver B has the key, but it is different. Keyserver B merges the keys, and sends the key back to Keyserver A. Keyserver A sees that the key is different, and merges the keys in its database, and sends the key back to Keyserver B. Keyserver B sees that the key is unchanged, and it does not write the key to the database or replog.

If a PGP Keyserver is offline, how and when is its database updated?

When a PGP Keyserver is offline, the Replication Engine on other PGP Keyservers continue to place new or modified keys for the target PGP Keyserver in the replog. When the PGP Keyserver is back on-line, the Replication Engines on the other PGP Keyservers automatically update the PGP Keyserver's database.

If the master PGP Keyserver is off-line, databases are not updated until the master PGP Keyserver is back online.

A LDAP Error Messages

This appendix lists the LDAP error messages found in the Access Log File. Each entry includes a brief description of the condition that generated the error:

LDAP Error Message	Description
0 (0x00) LDAP_SUCCESS	The request was successful.
1 (0x01) LDAP_OPERATIONS_ERROR	An unexpected Keyserver error was encountered. See the Error Log for more information.
2 (0x02) LDAP_PROTOCOL_ERROR	The client accessing the Keyserver is not following the proper protocol.
3 (0x03) LDAP_TIMELIMIT_EXCEEDED	The time limit for a single operation was exceeded. Only partial results were returned. If this occurs on a query operation, try again with a more restrictive query.
4 (0x04) LDAP_SIZELIMIT_EXCEEDED	The query operation matched more than the allowed number of entries. Only partial results were returned. Try the operation again with more restrictive query criteria.
5 (0x05) LDAP_COMPARE_FALSE	The comparison operation returned false.
6 (0x06) LDAP_COMPARE_TRUE	The comparison operation returned true.
7 (0x07) LDAP_STRONG_AUTH_NOT_SUPPORTED	Certain types of strong authentication are not supported.
8 (0x08) LDAP_STRONG_AUTH_REQUIRED	The operation performed requires a signed PGP request.
9 (0x09) LDAP_PARTIAL_RESULTS	This Keyserver could not satisfy the entire request. You may need to contact a referral Keyserver.
16 (0x10) LDAP_NO_SUCH_ATTRIBUTE	This requested attribute is not available.
17 (0x11) LDAP_UNDEFINED_TYPE	This error cannot be returned by the Keyserver.
18 (0x12) LDAP_INAPPROPRIATE_MATCHING	This error cannot be returned by the Keyserver.
19 (0x13) LDAP_CONSTRAINT_VIOLATION	Due to Keyserver policies, all User IDs and signatures were trimmed from the certificate. Nothing was left of the certificate to add.
20 (0x14) LDAP_TYPE_OR_VALUE_EXISTS	An attempt to add the same type or value was made.

LDAP Error Message	Description
21 (0x15) LDAP_INVALID_SYNTAX	An invalid keyblock was received by the Keyserver. See the Error Log for more details.
32 (0x20) LDAP_NO_SUCH_OBJECT	Attempted to reference an object that does not exist.
33 (0x21) LDAP_ALIAS_PROBLEM	This error cannot be returned by the Keyserver.
34 (0x22) LDAP_INVALID_DN_SYNTAX	The distinguished name does not have a valid syntax.
35 (0x23) LDAP_IS_LEAF	This error cannot be returned by the Keyserver.
36 (0x24) LDAP_ALIAS_DEREF_PROBLEM	This error cannot be returned by the Keyserver.
48 (0x30) LDAP_INAPPROPRIATE_AUTH	The authorization used for this request was invalid and unnecessary.
49 (0x31) LDAP_INVALID_CREDENTIALS	The certificate that you attempted to add does not contain the signatures required to pass policy. The certificate may have been placed in the pending bucket.
50 (0x32) LDAP_INSUFFICIENT_ACCESS	You do not have sufficient authority to perform the requested operation. The PGP signer of the request may not be authorized or the host may not have authority to the Keyserver.
51 (0x33) LDAP_BUSY	This error cannot be returned by the Keyserver.
52 (0x34) LDAP_UNAVAILABLE	The entry is not available or the PGP certificate has been disabled.
53 (0x35) LDAP_UNWILLING_TO_PERFORM	This operation is not supported.
54 (0x36) LDAP_LOOP_DETECT	This error cannot be returned by the Keyserver.
64 (0x40) LDAP_NAMING_VIOLATION	The submitted certificate did not match the certificate in the database. The certificate ID may have collided with the ID of another key.
65 (0x41) LDAP_OBJECT_CLASS_VIOLATION	This error cannot be returned by the Keyserver.
66 (0x42) LDAP_NOT_ALLOWED_ON_NONLEAF	An attempt to delete a non-leaf entry was made.
67 (0x43) LDAP_NOT_ALLOWED_ON_RDN	This error cannot be returned by the Keyserver.

LDAP Error Message	Description
68 (0x44) LDAP_ALREADY_EXISTS	The submitted certificate exists in the database and it has not changed. Or, a regular LDAP add was made and the entry already exists.
69 (0x45) LDAP_NO_OBJECT_CLASS_MODS	This error cannot be returned by the Keyserver.
70 (0x46) LDAP_RESULTS_TOO_LARGE	This error cannot be returned by the Keyserver.
80 (0x50) LDAP_OTHER	This error cannot be returned by the Keyserver.
81 (0x51) LDAP_SERVER_DOWN	The client detected that the Keyserver was not accessible. The host or port number may not be correct, the network may be having problems, or the Keyserver may be down.
82 (0x52) LDAP_LOCAL_ERROR	The client LDAP software had a problem.
83 (0x53) LDAP_ENCODING_ERROR	The data received by the Keyserver was incorrect or corrupted.
84 (0x54) LDAP_DECODING_ERROR	The data sent by the Keyserver was incorrect or corrupted.
85 (0x55) LDAP_TIMEOUT	This error cannot be returned by the Keyserver.
86 (0x56) LDAP_AUTH_UNKNOWN	This error cannot be returned by the Keyserver.
87 (0x57) LDAP_FILTER_ERROR	This error cannot be returned by the Keyserver.
88 (0x58) LDAP_USER_CANCELLED	The operation was aborted by the user.
89 (0x59) LDAP_PARAM_ERROR	The certificate received by the Keyserver is invalid. The keyblock may be missing.
90 (0x5a) LDAP_NO_MEMORY	A memory allocation problem occurred.

B Receiving Keys from MIT-Style Keyservers

This appendix describes how to configure a PGP Keyserver so that it can receive keys contained in a sync mail message from MIT-style keyservers.

The following instructions assume:

- The MIT-style keyserver is installed and running (in our examples, the server is installed in /MIT).
- The PGP Keyserver is installed in /opt/PGPkeysrv and is running on a UNIX machine, not necessarily the same machine on which the MIT-style keyserver is running.

This procedure consists of the following tasks:

- Configure the MIT-style keyserver to send mail messages to the PGP Keyserver.
- Configure the PGP Keyserver to receive mail messages from an MIT-style keyserver.
- Configure the mail program on the system where the PGP Keyserver resides to redirect the key messages to the PGP Keyserver.

To configure the MIT-style keyserver to send key messages to the PGP Keyserver:

1. Edit the configuration file, /MIT/etc/pksd.conf, of the sending MIT keyserver. Add the PGP Keyserver's email address as an argument to the sync site entry. For example, for a PGP Keyserver, specify sync site pgp-sync-keys@<host>.

   The name need not be pgp-sync-keys. After you enter the name, jot it down for future reference, as you must re-enter the name in one of the steps that follow.

To configure the PGP Keyserver to receive key messages from the MIT-style keyserver:

1. Edit the configuration file, /opt/PGPkeysrv/etc/pgpimportkey.conf, on the machine where the PGP Keyserver is running.
   - Add an entry for MailDir, the directory that will contain email messages that the PGP Keyserver receives from the MIT-style keyserver. Enter the absolute path to the directory, for example, /opt/PGPkeysrv/mail.
   - Add an entry for BinDir, the directory where the PGP Keyserver executables reside (pgpimport in particular). Enter the absolute path to the directory, for example, /opt/PGPkeysrv/bin.
   - Add an entry for SyncUrl, the URL for the PGP Keyserver, in the form ldap://<hostname>:<port>. The port number should match the listen port of the PGP Keyserver.

To configure the email program on the PGP Keyserver's machine to redirect the received key messages to the PGP Keyserver:

1. Configure the email program on the PGP Keyserver's system to redirect the mail addressed to the PGP Keyserver's sync site specified in /MIT/etc/pksd.conf to the PGP Keyserver. To do so, edit the /etc/aliases file on the machine where the PGP Keyserver is running (see below). Add the following entry:

   pgp-sync-keys: "|/opt/PGPkeysrv/bin/importkey.sh /opt/PGPkeysrv/etc/import-key.conf"

   The entire entry should appear on one line.

   pgp-sync-keys need not correspond to a real user.

   The alias for pgp-sync-keys pipes the mail message addressed to pgp-sync-keys to /opt/PGPkeysrv/bin/importkey.sh. This script invokes pgpimport to import the key from the message to the PGP Keyserver. It takes the shell script's configuration file, importkey.conf, as an argument.

2. Execute the UNIX command newaliases to process the changes you have made to /etc/aliases.

3. To ensure that mail and sendmail are working properly on the PGP Keyserver's machine, send ordinary email messages to and from the machine and verify that they are sent and received.

   The UNIX version of **pgpimport** has a **-d** command line option. Use of this option causes **pgpimport** to delete its keyfile argument (the received sync message from the MIT keyserver, in this example), after it is processed.

4. Change the permissions on the PGP Keyserver's /opt/PGPkeysrv/mail directory to allow the PGP Keyserver and the email program to read from and write to that directory.

C HTTP Support for PGP 5.0 Clients

PGP version 5.0 supports adding and searching keys from MIT-style keyservers. This style of keyserver is based on HTTP. The PGP Keyserver uses LDAP as a communication protocol between the client and the server. To allow existing PGP 5.0 clients to access the PGP Keyserver, an HTTP-to-LDAP gateway is included. The HTTP gateway consists of a series of CGI scripts that require access to a Web server.

In a UNIX environment, you may want to run the gateway on a different server from the PGP Keyserver to make your installation more secure or flexible.

The bundled Web server is preconfigured to support an HTTP gateway. To disable the HTTP gateway, use the configuration wizard. To limit accessibility, you can manually modify the httpd.conf file.

D Keyserver and Engine Command Line Switches

This appendix lists the PGP Keyserver and Replication Engine command line switches.

PGP Keyserver command line switches

The following table describes each of the PGP Keyserver command line switches.

Keyserver Command Line Switch	Description
-a	Instructs the PGP Keyserver to assume that all signatures have already passed the policy requirements. All other policy checks are enforced. Use this option to copy a large number of verified keys from one PGP Keyserver to another.
-c	Checks the current configuration file for accuracy. Does not start the PGP Keyserver. When you make configuration changes, use this option to verify that the new configuration values are valid.
-f <*file*>	Identifies the configuration file that the PGP Keyserver uses. If -f is not specified, uses default found in the Windows registry (configured with the configuration wizard) or ../etc/pgpcertd.conf or .cfg.
-g	Windows. Tells the software to start as a command line PGP Keyserver rather than an NT service. Required if the PGP Keyserver is invoked on the command line.
-p <*port*>	Identifies the port number that the PGP Keyserver listens to for client requests and certificate submittals. Defaults to the port number listed in the configuration file or port 389.
-V	Displays the program's copyright and version number. When this option is used, the program prints version information then exits.

Keyserver Command Line Switch	Description
-d *<level>*	Turns on the debug mode and provides a level of information based on the level you select. The following are the debug levels: Debug Level — Description 1 — Prevent process from detaching from terminal (UNIX) 2 — All normally logged event messages, as if 'LogLevel verbose' was set. 4 — Function trace 8 — LDAP arguments 16 — Shows active threads 32 — LDAP filter display 64 — Configuration file statistics 128 — ACL statistics 256 — LDIF parsing errors 512 — More detailed error conditions 1024 — More detailed processing messages 2048 — Info conditions 8192 — Misc messages 65535 — All of the above debug levels combined. This switch is primarily used for debugging purposes. Do not use this switch unless you are very familiar with this process or you are consulting with a Technical Support Engineer.
-s	When used in conjunction with **SecureMode** set to Optional, allows you to run the PGP Keyserver automatically from a script (PGP Keyserver will start unattended with LDAPS disabled). If you do not use the **-s** option and LDAPS is enabled, the PGP Keyserver will prompt for a passphrase, if needed.
-t <*port*>	Identifies the port number that the PGP Keyserver listens to for Secure Mode. Defaults to port 636. If **-t** is not present, the PGP Keyserver uses the value for **SecurePort** found in the configuration file. **-t -1** (minus one) disables LDAPS and Secure Mode.

Replication Engine command line switches

Engine command line switches	Description
-f *<file>*	Identifies the configuration file that you want the Replication Engine to use. By default, the Replication Engine uses pgpcertd.cfg (in UNIX, pgpcertd.conf*)*, in the ../etc/ directory. This file contains all of the configuration settings that affect the Replication Engine.
-t *<directory>*	Identifies the directory for the Replication Engine's temporary files. This option overrides the TempPath configuration parameter. For more information, see "Temporary File Path - TempPath <path>" on page 93.
-r *<file>*	Identifies the log file you want the Replication Engine to use. This option overrides the RepLogFile value in the configuration file.
-o	During normal operation, the Replication Engine continuously monitors the replication log file for new entries. When you use this option, the Replication Engine looks at the replication log file only once. After processing all the entries, it then exits.
-c	Checks the current configuration file for accuracy. When you make configuration changes, use this option to verify that the new configuration values are valid. UNIX. Temporarily starts and stops the Replication Engine. Configuration warnings and error messages are sent to the standard error device (stderr). Windows. Does not start the Replication Engine.
-d *<level>*	Turns on debug mode and gives you information based on the level you choose. The levels are the same as those for the PGP Keyserver.
-g	Windows. Required if the Engine is invoked from the command line. Indicates that the Engine is not running as an NT service.
-V	Displays the program's copyright and version number. When this option is used, the program prints version information then exits.

The temporary files used by the Replication Engine can become quite large. Make sure they are stored on a partition that is large enough to hold this data. Use the **-t** option or the **TempPath** configuration setting to explicitly designate where these temporary files are stored.

Glossary

Additional Decryption Key (ADK)
Enables a company to access information encrypted by its employees in the event of an emergency. Any information encrypted to the user's key is also encrypted to the ADK. When someone inside or outside the organization encrypts information to a user, the information is also encrypted to the ADK.

ASCII-armored text
Binary information that has been encoded using a standard, printable, 7-bit ASCII character set, for convenience in transporting the information through communication systems. In the PGP program, ASCII armored text files are given the default filename extension, and they are encoded and decoded in the ASCII radix-64 format.

authentication
The determination of the origin of encrypted information through the verification of someone's digital signature or someone's public key by checking its unique fingerprint.

certificate
A unique digital code used to encrypt, sign, decrypt, and verify email messages and files. In traditional PGP parlance, certificates are generally referred to as keys.

certify
To sign another person's public key.

Certification Authority (CA)
One or more trusted individuals who are assigned the responsibility of certifying the origin of keys and adding them to a common database.

conventional encryption
Encryption that relies on a common passphrase instead of public key cryptography. The file is encrypted using a session key, which encrypts using a passphrase that you will be asked to choose

decryption
A method of unscrambling encrypted information so that it becomes legible again. The recipient's private key is used for decryption.

digital signature
See signature.

encryption
A method of scrambling information to render it unreadable to anyone except the intended recipient, who must decrypt it to read it.

fingerprint
A uniquely identifying string of numbers and characters used to authenticate public keys. This is the primary means for checking the authenticity of a key.

gateway
A device that connects networks using different communications protocols so that information can be passed from one to the other.

host
A computer that provides access to other computers.

introducer
A person or organization who is allowed to vouch for the authenticity of someone's public key. You designate an introducer by signing their public key.

key
A digital code used to encrypt and sign and decrypt and verify email messages and files. Keys come in keypairs and are stored on keyrings.

key escrow
A practice where a user of a public key encryption system surrenders their private key to a third party thus permitting them to monitor encrypted communications.

key fingerprint
A uniquely identifying string of numbers and characters used to authenticate public keys. For example, you can telephone the owner of a public key and have him or her read the fingerprint associated with their key so you can compare it with the fingerprint on your copy of their public key to see if they match. If the fingerprint does not match, then you know you have a bogus key.

KeyID
A legible code that uniquely identifies a keypair. Two keypairs may have the same user ID, but they will have different KeyIDs.

keypair
A public key and its complimentary private key. In public-key cryptosystems, like the PGP program, each user has at least one keypair.

key reconstruction
Key reconstruction is used to re-create lost keys. When users generate a new key, they can create a set of simple questions and answers that are easy for them to remember and very difficult for anyone else to guess. The pre-defined questions are good examples of questions that work—they are very general, but they produce very personal answers. Users can also can also customize questions. If a key is lost or passphrase is forgotten, the information can be re-created if the user provides correct answers to the set of questions.

keyring
A set of keys. Each user has two types of keyrings: a private keyring and a public keyring.

keyserver
A repository for public keys and certificates.

LDAP
An acronym for the Lightweight Directory Access Protocol which specifies how directory services are provided through a standard query interface.

message digest
A compact "distillate" of your message or file checksum. It represents your message, such that if the message were altered in any way, a different message digest would be computed from it

meta-introducer
A trusted introducer of trusted introducers.

NTFS
Acronym for NT file system. An advanced file system designed for use specifically with the Windows NT operating system.

passphrase
A series of keystrokes that allow exclusive access to your private key which you use to sign and decrypt email messages and file attachments.

plaintext
Normal, legible, un-encrypted, unsigned text.

private key
The secret portion of a keypair-used to sign and decrypt information. A user's private key should be kept secret, known only to the user.

private keyring
A set of one or more private keys, all of which belong to the owner of the private keyring.

public key
One of two keys in a keypair-used to encrypt information and verify signatures. A user's public key can be widely disseminated to colleagues or strangers. Knowing a person's public key does not help anyone discover the corresponding private key.

public keyring
A set of public keys. Your public keyring includes your own public key(s).

public-key cryptography
Cryptography in which a public and private keypair is used, and no security is needed in the channel itself.

remote authentication
Your machine can use PGP keys and X.509 certificates to identify itself to other machines.

sign
To apply a signature.

signature
A digital code created with a private key. Signatures allow authentication of information by the process of signature verification. When you sign a message or file, the PGP program uses your private key to create a digital code that is unique to both the contents of the message and your private key. Anyone can use your public key to verify your signature.

subnet
In general, a network that forms part of a larger network.

text
Standard, printable, 7-bit ASCII text.

trusted
A public key is said to be trusted by you if it has been certified by you or by someone you have designated as an introducer.

trusted introducer
Someone who you trust to provide you with keys that are valid. When a trusted introducer signs another person's key, you trust that their keys are valid, and you do not need to verify their keys before using them.

unmount
To make a physical disk inaccessible to a computer's file system.

user ID
A text phrase that identifies a keypair. For example, one common format for a user ID is the owner's name and email address. The user ID helps users (both the owner and colleagues) identify the owner of the keypair.

verification
The act of comparing a signature created with a private key to its public key. Verification proves that the information was actually sent by the signer, and that the message has not been subsequently altered by anyone else.

web of trust
A distributed trust model used by PGP to validate the ownership of a public key where the level of trust is cumulative, based on the individuals' knowledge of the introducers.

Index

Symbols

Numerics

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W