

PGP Certificate Server for NT Administrator's Guide

Version 2.5

COPYRIGHT

Copyright © 1999 Network Associates Technology, Inc. All Rights Reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form or by any means without the written permission of Network Associates Technology, Inc., or its suppliers or affiliate companies.

PGP Certificate Server for NT, Version 2.5.1

6-99. Printed in the United States of America.

TRADEMARK ATTRIBUTIONS

** ActiveHelp, Bomb Shelter, Building a World of Trust, CipherLink, Clean-Up, Cloaking, Compass 7, CNX, CyberCop, CyberMedia, Data Security Letter, Discover, Distributed Sniffer System, Dr Solomon's, Enterprise Secure Cast, First Aid, ForceField, Gauntlet, GMT, Hunter, ISDN Tel/Scope, LM 1, LANGuru, Leading Help Desk Technology, Magic Solutions, MagicSpy, MagicTree, Magic University, MagicWin, MagicWord, McAfee Associates, McAfee, MoneyMagic, More Power To You, Multimedia Cloaking, NetCrypto, NetRoom, NetScan, Net Shield, NetShield, NetStalker, Net Tools, NetOctopus, NetStalker, Network Associates, Network General, Network Uptime!, NetXRay, Nuts & Bolts, PC Medic, PCNotary, PGP, PGP (Pretty Good Privacy), PocketScope, PowerTelnet, Pretty Good Privacy, PrimeSupport, RecoverKey, RecoverKey-International, ReportMagic, RingFence, Router PM, Safe & Sound, SalesMagic, SecureCast, Service Level Manager, ServiceMagic, Site Meter, Sniffer, SniffMaster, SniffNet, Stalker, Statistical Information Retrieval (SIR), SupportMagic, T-POD, TeleSniffer, TIS, TMach, TMeg, Trusted Mach, Trusted Mail, Total Network Visibility, Total Virus Defense, Uninstaller, Virex, Virex-PC, Virus Forum, ViruScan, VirusScan, VShield, WebScan, WebShield, WebSniffer, WebStalker WebWall, and ZAC 2000* are registered trademarks of Network Associates and/or its affiliates in the US and/or other countries. All other registered and unregistered trademarks in this document are the sole property of their respective owners.

Portions of this software may use public key algorithms described in U.S. Patent numbers 4,200,770, 4,218,582, 4,405,829, and 4,424,414, licensed exclusively by Public Key Partners; the IDEA(tm) cryptographic cipher described in U.S. patent number 5,214,703, licensed from Ascom Tech AG; and the Northern Telecom Ltd., CAST Encryption Algorithm, licensed from Northern Telecom, Ltd. IDEA is a trademark of Ascom Tech AG. Network Associates Inc. 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. The compression code in PGP is by Mark Adler and Jean-Loup Gailly, used with permission from the free Info-ZIP implementation. LDAP software provided courtesy University of Michigan at Ann Arbor, Copyright © 1992-1996 Regents of the University of Michigan. All rights reserved. This product includes software developed by the Apache Group for use in the Apache HTTP server project (<http://www.apache.org/>). Copyright © 1995-1999 The Apache Group. All rights reserved. See text files included with the software or the PGP web site for further information.

LICENSE AGREEMENT

NOTICE TO ALL USERS: FOR THE SPECIFIC TERMS OF YOUR LICENSE TO USE THE SOFTWARE THAT THIS DOCUMENTATION DESCRIBES, CONSULT THE README.1ST, LICENSE.TXT, OR OTHER LICENSE DOCUMENT THAT ACCOMPANIES YOUR SOFTWARE, EITHER AS A TEXT FILE OR AS PART OF THE SOFTWARE PACKAGING. IF YOU DO NOT AGREE TO ALL OF THE TERMS SET FORTH THEREIN, DO NOT INSTALL

THE SOFTWARE. IF APPLICABLE, YOU MAY RETURN THE PRODUCT TO THE PLACE OF PURCHASE FOR A FULL REFUND.

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.

Network Associates, Inc.

(408) 988-3832 main

3965 Freedom Circle

Santa Clara, CA 95054

<http://www.nai.com>

info@nai.com

* is sometimes used instead of the © for registered trademarks to protect marks registered outside of the U.S.

LIMITED WARRANTY

Limited Warranty. Network Associates warrants that for sixty (60) days from the date of original purchase the media (for example diskettes) on which the Software is contained will be free from defects in materials and workmanship.

Customer Remedies. Network Associates' and its suppliers' entire liability and your exclusive remedy shall be, at Network Associates' option, either (i) return of the purchase price paid for the license, if any, or (ii) replacement of the defective media in which the Software is contained with a copy on nondefective media. You must return the defective media to Network Associates at your expense with a copy of your receipt. This limited warranty is void if the defect has resulted from accident, abuse, or misapplication. Any replacement media will be warranted for the remainder of the original warranty period. Outside the United States, this remedy is not available to the extent Network Associates is subject to restrictions under United States export control laws and regulations.

Warranty Disclaimer. To the maximum extent permitted by applicable law, and except for the limited warranty set forth herein, THE SOFTWARE IS PROVIDED ON AN "AS IS" BASIS WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. WITHOUT LIMITING THE FOREGOING PROVISIONS, YOU ASSUME RESPONSIBILITY FOR SELECTING THE SOFTWARE TO ACHIEVE YOUR INTENDED RESULTS, AND FOR THE INSTALLATION OF, USE OF, AND RESULTS OBTAINED FROM THE SOFTWARE. WITHOUT LIMITING THE FOREGOING PROVISIONS, NETWORK ASSOCIATES MAKES NO WARRANTY THAT THE SOFTWARE WILL BE ERROR-FREE OR FREE FROM INTERRUPTIONS OR OTHER FAILURES OR THAT THE SOFTWARE WILL MEET YOUR REQUIREMENTS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, NETWORK ASSOCIATES DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT WITH RESPECT TO THE SOFTWARE AND THE ACCOMPANYING DOCUMENTATION. SOME STATES AND JURISDICTIONS DO NOT ALLOW LIMITATIONS ON IMPLIED WARRANTIES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. The foregoing provisions shall be enforceable to the maximum extent permitted by applicable law.

Table of Contents

Preface	xi
Who Should Read This Guide	xi
What's in This Guide	xi
Conventions Used in this Guide	xii
Typographical Conventions	xii
Special Advisements	xii
For More Information	xii
Customer Service	xiii
Technical Support	xiii
Year 2000 Compliance	xiv
Your Feedback is Welcome	xiv
Related Reading	xiv
Chapter 1. The PGP Certificate Server	17
General Features	17
How it Works	18
What is a Certificate?	18
Installation and Configuration	18
Operation	19
Submitting Keys	19
Retrieving Keys	20
Importing and Exporting Keys	20
Replication of the Database to Other Servers	21
Monitoring usage and activity	21
Getting Started	22
Installation and operation	22
Launching the Server as an NT service from a local console	23
Using the Server's console	24
Launching the Server as an NT Service from a remote console	24
Secure Mode	26
Troubleshooting	26

Chapter 2. Configuration	27
Configuring the Server	27
Required parameters for the Server	27
Configuring the Server's console	28
Required Server parameters for console connection	28
Overriding settings in the configuration file	29
Extracting Key IDs for configuration purposes	29
Verifying validity of new configuration settings	30
Setting up the Configuration/Monitoring wizard	30
Using the Configuration/Monitoring wizard	31
Examining and editing the configuration file	32
General configuration settings	35
CommandPort <Port> — Command Port	35
CommandTimeout <Seconds> — Command Timeout	36
CycleLogDay <frequency> — Day to Cycle Log	36
CycleLogKeep <number> — Logs to Keep	36
CycleLogTime <time> — Time to Cycle Log	36
LogLevel <level> — Logging Level	37
Port <Port> — Port	37
SizeLimit <size> — Size Limit	37
TimeLimit <seconds> — Time Limit	37
Database Configuration Settings	38
CacheEntries <number of entries> — Cache Entries	38
DBCacheSize <size> — DB Cache Size	38
Directory <path> — Database Directory	38
IdleSyncTimeout <seconds> — Idle Sync Timeout	38
Mode <file permissions> — Database File Permissions	38
ReadOnly on off — Database Access Mode	38
Certificate policy configuration settings	39
AllowSigID <keyID> — Allowed Certificate Signatures	39
MustSigID <keyID> — Required Certificate Signatures	39
PolicyFailures pending error — Action on Key Policy Failure ..	40
TrimPhotoIDs yes no — Remove PhotoIDs	40
TrimSigs yes no — Remove Unallowed Signatures	40
TrimUsers yes no — Remove Unallowed User IDs	40

Certificate policy configuration matrix41
Replication Engine configuration settings41
Allow keyid <KeyID> command42
RepdCommandPort <command port> — Replication command port42
RepdCommandTimeout <seconds> — Replication command timeout42
Replica [<protocol>://] <hostname> or <IPaddress> [:<port>] — Hosts to Replicate Database to43
ReplicationSecureKeyID <KeyID> — Replication Secure Key ID	.43
RepLogFile <filename> — Replication Log File43
Secure mode configuration settings44
PrivateKeyRing <filename> — Private Key Ring44
PublicKeyRing <filename> — Public Key Ring44
RandSeedFile <filename> — Random seed file45
SecureMode <Mode> — Secure Mode45
SecurePort <Port> — Secure Port45
ServerSecureKeyID <KeyID> — Server Secure Key ID45
Access configuration settings46
AccessLogDetails <type> — Items to Log in Access Log47
AccessLogFile <filename> — Access Log File47
Allow keyid <KeyID> command47
Allow <who> <access> — Allow access by47
DefaultAccess none compare search read add delete all — Default Access50
Chapter 3. Operation53
Starting the Server's console53
Starting the Server's console from the Start menu53
Starting the Server's console from the command line54
Starting the Certificate Server55
Starting the Certificate Server as an NT service from the console56
Stopping the NT service57
Starting the Server from the command line57
Configuring the Server to start automatically60
Invoking the Server with the -s auto-start option60
Verifying that the Server is running60

Running multiple Servers on the same machine60
Starting the Replication Engine's console61
Preparing to start the Replication Engine62
Starting the Replication Engine from the console64
Engine authentication64
Starting the Replication Engine from the command line64
Verifying that the Replication Engine is running66
Running multiple engines on the same machine66
Stopping the Replication Engine from the console66
Chapter 4. Using the Server67
Learning about Servers67
Server and console compatibility67
Server, PGPkeys, and the console67
Console retrieves Monitor and Event data67
Server times out and closes connection67
Server authentication68
Configuring the Server to start automatically68
HTTP support for PGP 5.0 clients68
Configuring Internet Information Server 3 for use as an HTTP Gateway70
Administering the Server71
Resolving keys in the pending bucket72
Importing and exporting keys72
Importing keys to the Certificate Server72
Exporting keys from the Certificate Server72
Using the Server's console73
The Control panel74
The Command panel76
The Monitor and Events panels77

Chapter 5. Using the Replication Engine	79
Server configurations80
Server configuration models80
Master slave model (high efficiency, medium tolerance)80
Star model (medium efficiency, medium tolerance)81
Ring model (high efficiency, low tolerance)82
Fully connected model (low efficiency, high tolerance)83
Examples of different server configurations84
How the replication process works85
When does replication occur?86
How and where is the replication log (repllog) maintained?86
Can two Servers replicate to each other?87
If a Server is off-line, how and when is its database updated?87
Adding a new Server88
Learning about the Replication Engine88
Engine and console compatibility88
Console retrieves Monitor and Event data89
Engine times out and closes connection89
Using the Replication Engine's console89
The Engine console's Control panel90
The Engine console's Command panel91
NT Service fields91
Name or IP Address and Command Port92
Command Timeout92
The Engine console's Monitor and Events panel92
Chapter 6. Secure Mode	93
Setting Up Secure Mode93
Using Secure Mode97
Chapter 7. Monitoring and Logging	99
Monitoring Operations99
Monitoring the Server99
Events Panel Not Displaying Information100
Monitoring the Replication Engine100

Monitoring Certificate Server Activity	101
Statistics	101
Uptime	101
Ops Completed	101
Current Connections	102
Total Connections	102
Bytes Transmitted	102
Entries Served	102
Client List	102
Monitoring Certificate Server Events and Errors	102
Monitoring Replication Engine Activity	103
Status	103
Queue Size	103
Last Update Time	103
Total Replications	104
Monitoring the Replication Engine's Events and Errors	104
Monitoring Remotely with the Wizard	105
Examining the Access Log File	106
Sample Access Log File Entries	108
Access Log File Cycling	108
Naming Convention for Cycled Files	108
Retention Period for Cycled Files	109
Examining the NT Event Log	109
LDAP Error Messages	109
Removing the Software	113
Glossary	115
Index	119

Preface

Who Should Read This Guide

This guide describes how to install, configure, operate, and maintain a PGP Certificate Server. The guide is for System Administrators or others who are responsible for setting up and running the Server. The Certificate Server allows PGP users to submit and retrieve keys according to the policies enforced at your site.

What's in This Guide

Chapter 1 *The PGP Certificate Server*

Describes the PGP Certificate Server's features and explains how the Server works.

Chapter 2 *Configuration*

Describes how to configure the PGP Certificate Server.

Chapter 3 *Operation*

Describes how to run the Server and Replication Engine.

Chapter 4 *Using the Replication Engine*

Describes how to use the Replication Engine.

Chapter 5 *Using the Server*

Describes how to use the Server.

Chapter 6 *Secure Mode*

Describes how to set up and use Secure Mode.

Chapter 7 *Monitoring and Logging*

Describes how to monitor Server usage and how to interpret the log files.

Conventions Used in this Guide

The following sections explain the conventions used in this manual to delineate and emphasize important terms, concepts, and instructions.


Typographical Conventions

New terms, variables, commands, and code samples appear in a different style or font to help distinguish them from the surrounding text.


- New terms are shown in *italics* and are generally defined in context or, if necessary, are elaborated on in greater detail in the Glossary. Variables are also shown in *italics*, for example, `http://<www.company.com>/certserver/default.htm`
- Commands are shown in **bold** to indicate information that appears on the screen.
- Code samples are shown in Courier font (this is an example of Courier).

Special Advisements

The following special advisements are used to call your attention to information that requires consideration.

 **NOTE:** Notes give supplemental information that emphasizes a concept or explains a caveat regarding the current topic of discussion.

TIP: Tips give specific guidelines you should follow or precautions you should take when carrying out a specific task.

 **WARNING:** Alerts are warnings about conditions or procedures that could result in unwanted consequences unless specific measures are observed.

For More Information

There are several ways to find out more about Network Associates and its products.

Customer Service

To order products or obtain product information, contact the Network Associates Customer Care department.

You can contact Customer Care at one of the following numbers Monday through Friday between 6:00 A.M. and 6:00 P.M. Pacific time.

Phone (408) 988-3832

Fax (408) 970-9727

Or write to:

Network Associates, Inc.
3965 Freedom Circle
Santa Clara, CA 95054
U.S.A.

Technical Support

Network Associates is famous for its dedication to customer satisfaction. We have continued this tradition by making our site on the World Wide Web a valuable resource for answers to technical support issues. We encourage you to make this your first stop for answers to frequently asked questions, for updates to Network Associates software, and for access to Network Associates news and encryption information.

World Wide Web <http://www.nai.com>

Technical Support for your PGP product is also available through these channels:

Phone (970) 522-2952

Fax (408) 970-9727

Email PGPSupport@pgp.com

To provide the answers you need quickly and efficiently, the Network Associates technical support staff needs some information about your computer and your software. Please have this information ready before you call:

- PGP product name
- PGP product version
- Computer platform and CPU type

- Amount of available memory (RAM)
- Operating system and version and type of network
- Content of any status or error message displayed on screen, or appearing in a log file (not all products produce log files)
- Email application and version (if the problem involves using PGP with an email product, for example, the Eudora plug-in)

Year 2000 Compliance

Information regarding NAI products that are Year 2000 compliant and its Year 2000 standards and testing models may be obtained from NAI's website at <http://www.nai.com/y2k>. For further information, email y2k@nai.com.

Your Feedback is Welcome

We continually improve our product documentation and welcome customer feedback. If you would like to provide input, please send email to us at the following address:

tns_documentation@nai.com

Related Reading

Here are some documents that you may find helpful in understanding cryptography:

Non-Technical and beginning technical books

- “*Cryptography for the Internet*,” by Philip R. Zimmermann. Scientific American, October 1998. This article, written by PGP's creator, is a tutorial on various cryptographic protocols and algorithms, many of which happen to be used by PGP.
- “*Privacy on the Line*,” by Whitfield Diffie and Susan Eva Landau. MIT Press; ISBN: 0262041677. 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, and contains information that even a lot of experts don't know.
- “*The Codebreakers*,” by David Kahn. Scribner; ISBN: 0684831309. 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, and published a revised edition in 1996. This book won't teach you anything about how cryptography is accomplished, but it has been the inspiration of the whole modern generation of cryptographers.

- “*Network Security: Private Communication in a Public World*,” by Charlie Kaufman, Radia Perlman, and Mike Spencer. Prentice Hall; ISBN: 0-13-061466-1. This is a good description of network security systems and protocols, including descriptions of what works, what doesn't work, and why. Published in 1995, it doesn't have many of the latest technological advances, but is still a good book. It also contains one of the most clear descriptions of how DES works of any book written.

Intermediate books

- “*Applied Cryptography: Protocols, Algorithms, and Source Code in C*,” by Bruce Schneier, John Wiley & Sons; ISBN: 0-471-12845-7. This is a good beginning technical book on how a lot of cryptography works. If you want to become an expert, this is the place to start.
- “*Handbook of Applied Cryptography*,” by Alfred J. Menezes, Paul C. van Oorschot, and Scott Vanstone. CRC Press; ISBN: 0-8493-8523-7. This is the technical book you should read after Schneier's book. There is a lot of heavy-duty math in this book, but it is nonetheless usable for those who do not understand the math.
- “*Internet Cryptography*,” by Richard E. Smith. Addison-Wesley Pub Co; ISBN: 0201924803. This book describes how many Internet security protocols work. Most importantly, it describes how systems that are designed well nonetheless end up with flaws through careless operation. This book is light on math, and heavy on practical information.
- “*Firewalls and Internet Security: Repelling the Wily Hacker*,” by William R. Cheswick and Steven M. Bellovin. Addison-Wesley Pub Co; ISBN: 0201633574. This book is written by two senior researchers at AT&T Bell Labs and is about their experiences maintaining and redesigning AT&T's Internet connection. Very readable.

Advanced books

- “*A Course in Number Theory and Cryptography*,” by Neal Koblitz. Springer-Verlag; ISBN: 0-387-94293-9. An excellent graduate-level mathematics textbook on number theory and cryptography.
- “*Differential Cryptanalysis of the Data Encryption Standard*,” by Eli Biham and Adi Shamir. Springer-Verlag; ISBN: 0-387-97930-1. This book describes the technique of differential cryptanalysis as applied to DES. It is an excellent book for learning about this technique.

This chapter describes the PGP Certificate Server's features and explains how the Server works. The last section, "[Getting Started](#)" on page 22, includes complete instructions to install, configure, and run a Certificate Server.

You must have administrative privileges on the machine on which the PGP Certificate Server is running. To remotely start the NT service from the console, you also need domain administrative privileges. This guide assumes that you are the System Administrator.

-
- ❏ **NOTE:** Throughout this documentation we refer to the PGP Certificate Server as the Server and the Replication Engine as the engine.
-

General Features

The PGP Certificate Server allows users to submit and retrieve *keys* from a database. A key is a digital code used in conjunction with a cryptographic algorithm to encrypt, sign, decrypt, and verify email messages and files. Information encrypts differently with different keys. The Server uses a set of user-defined policies to control key submission and retrieval.

Server features include the following:

- Automated installation and configuration of the Server through easy-to-use scripts and a Web-based **Configuration/Monitoring wizard**.
- Single point-of-control user interface to start and stop the Server and monitor other activities.
- Flexible key retrieval that supports searches on multiple key attributes, such as the key type, key ID, creation date, and so on.
- Authentication safeguards that limit access to restricted Server functions (includes access controls and signature verification).
- *PGP Replication Engine* that allows you to replicate database entries to multiple Servers. The databases on these Servers are automatically updated to reflect the contents of the database on the primary Server.

How it Works

The PGP Certificate Server is designed to run on the Windows NT platform. The Server 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. The Server includes a Replication Engine to propagate the contents of the master database to multiple Servers, if required.

The Server enforces the certificate policy established during configuration. The certificate policy identifies the criteria that the Server uses to enforce the acceptance or rejection of keys. The certificate policy also identifies how keys are retrieved by users.

Some older versions of PGP only support access to key servers over the Web. For these versions, the Server includes a CGI interface that supports the HTTP protocol. This also allows PGP clients to access the Server through firewalls without an LDAP proxy.

What is a Certificate?

A digital *certificate*, called a certificate throughout this document, is information included with a person's public key that helps others verify that a key is genuine or valid. A digital certificate consists of three things:

- A public key.
- Certificate information (identifies information about the user, such as name, user ID, and so on).
- A digital signature.

Installation and Configuration

Installation of the Server is performed using a simple-to-use *installation wizard*. The wizard makes sure all the necessary software components are loaded in the proper sequence and stored in the appropriate directories.

Note that the different components (Certificate Server, consoles, and Replication Engine) can be installed on different machines. You may choose to install the Certificate Server as an NT service on one machine, the Certificate Server's console on another machine, and the Replication Engine and Replication Engine's console on a third machine. See the *PGP Installation Guide* for installation instructions.

The Replication Engine usually resides on the same machine as one of the Certificate Servers. Replica or slave Certificate Servers can be on different machines. The Replication Engine's console can reside on a different machine than the Replication Engine.

Configuration of the Server is performed using a Web-based **Configuration/Monitoring wizard**. The wizard helps you set up the Server to meet the requirements of your site. This configuration method should address the needs of most installations. However, if you need to change the Server's configuration, you can edit the Server's configuration file. Some configuration is allowed from the console as well.

Operation

You control the Server's major functions (for example, starting and stopping the Server) from the Server's graphical console. The console, which may not be on the same machine as the Server, also allows you to monitor Server activity.

When the Server is running, it responds to user requests to add, search for, and retrieve keys. The Server uses two sets of criteria to accept or reject keys: user access level and configuration parameters in the configuration file.

Submitting Keys

When a key is submitted to the Server, the Server checks to see if the key passes the policy requirements established during configuration (for more information about this criteria, see [“Certificate policy configuration settings” on page 39](#)).

- The Server verifies that the key is signed by the required entities, as identified during Server configuration.
- The Server verifies the Authorized signatures on the key, as identified during Server configuration. The Server removes all unauthorized signatures or User IDs from the key before storing the key in the Server's database.

After enforcing the policy requirements, the Server 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 Server (for more information about the pending bucket, see [“Administering the Server” on page 71](#)).

Retrieving Keys

When a key is placed on the Server, 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 interface to access keys. As System Administrator, your authority level, established during Server configuration, allows you to add, disable, and delete keys from the Server.

For more information on how to configure these settings, see [Chapter 2, "Configuration."](#)

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 Server 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.) You can also export keys to any client machine from the machine running the Server.

Replication of the Database to Other Servers

The Replication Engine is a robust replication mechanism used to propagate the contents of a primary or master Server's database to one or more slave Servers. A replication daemon monitors the master Server and updates the slave Servers' databases whenever a change occurs on the master Server.

You identify the slave Servers when you run the **Configuration/Monitoring wizard**. The wizard stores this information in the master Server's configuration file.

Not all installations will use the master-slave Server configuration. A variety of Server configuration models are described in [Chapter 5, "Using the Replication Engine."](#)

Monitoring usage and activity

The statistics collected by the Server and the Replication Engine allow you to monitor usage and track various activities. During configuration you identify the types of activities that you want to track and the level of detail you want the Server to record.

There are several ways to find out how the Server and Replication Engine are performing:

- Monitor the activity in real-time (click the Monitor tab from the console or use the **Configuration/Monitoring wizard's** monitoring features). Use the wizard when you do not have direct access to the machine where the Server resides.
- Consult the *Access Log File* (stores a more complete record of activities). This file is available from the **Configuration/Monitoring wizard**.
- Check the *Events panel* (displays any errors that occur while the Server or Replication Engine is running).
- Consult the *NT Event Log* (lists all of the errors that have occurred over a longer period of time).

Getting Started

-
- **NOTE:** You must have administrative privileges on the machine on which the PGP Certificate Server (Server) or Replication Engine (engine) is running. To remotely start the Server or engine as an NT service from the console, you also need administrative privileges on the domains of both machines. This guide assumes that you are the System Administrator.
-

There are two major components that you can install: the Certificate Server (Server) and the Replication Engine (engine).

- The Server allows users to submit and retrieve keys from a database. The Server uses a set of user-defined policies to control key submission and retrieval.
- The engine allows you to replicate database entries on a primary Server to multiple secondary Servers. The databases on these Servers are automatically updated to reflect the contents of the database on the primary Server.

The Server and engine each have a console. In addition, the Server has a **Configuration/Monitoring Wizard** that requires a web server.

The different components (Server, Server console, engine, and Engine console) can be installed on different machines. The only exception to this rule is that the primary server and engine must be on the same machine.

The following is a brief outline of the steps you need to take to install the Server and engine and get them up and running.

Installation and operation

The following scenario assumes that the Certificate Server and console are installed on the same machine.

1. Run the PGP setup program to install the Certificate Server and Replication Engine. Specific instructions are included in the *PGP Installation Guide*. The components that you can install are:
 - PGP Certificate Server
 - PGP Replication Engine
 - PGP Certificate Server Console
 - PGP Replication Engine Console
 - Certificate Server *Administrator's Guide*
 - Certificate Server Web Interface

Launching the Server as an NT service from a local console

1. To start the Server as an NT service on the local machine, start the Server console (**Start -> Programs -> PGP Certificate Server -> PGP Certificate Server console**).

If you have multiple keys on your default keyring, you are prompted to select the console's TLS key. If you have only one key on that keyring, it is automatically selected.

If the selected key requires a passphrase, you are prompted to enter the passphrase.

2. Verify that the NT Service **Disabled** checkbox on the console's **Command** panel is not checked. The console cannot launch the Server as an NT service if this box is checked.
3. Verify that the default **Command Port** displayed on the console's **Command** panel is not used by another process on the local machine. If it is, enter a different port number.
4. Verify that the default **Configuration File** displayed on the console's **Control** panel is the file that you want to use. If it is not, enter the correct file.
5. Click **Create Database** on the Server console's **Control** panel. The Server displays a dialog box when it has successfully created a database.
6. Click **Start** on the Server console's **Control** panel to start the Server as an NT service. The Server will prompt for a TLS key and passphrase if a passphraseless key has not been specified in the configuration file and the default configuration file has been modified to require Secure Mode. To allow only local connections to the Server and disable the secure port, press **Cancel**. Note that the **Start** button becomes a **Stop** button.
7. The Server continues to run as an NT service even if the console is closed and even if you, the current user, log off the NT workstation.
8. To stop the NT service, click **Stop**.

Using the Server's console

Use the panels on the Server's console to perform the following tasks:

Control panel - Identify the Server's port numbers and configuration file, create the Server's initial database, tell the Server not to perform signature verification, check the Server's configuration file for validity, start, stop, and restart the Server.

Command panel - Use the controls on this panel to define information about the Server running as an NT service or as a command line application on the local or remote machine.

Monitor panel - The Monitor panel gives information about the keys submitted and retrieved from the Server. Use the **Auto Update** option on this panel to update this information automatically.

Events panel - The Events panel displays all Server events and errors. When the Events panel is selected, it displays all of the messages generated since the Server was last started. Click **Refresh** to update the listing.

For specific information on the **Control** and **Command** panels, see [Chapter 4, "Using the Server."](#) For specific information on the Monitor and Events panels, see [Chapter 7, "Monitoring and Logging."](#)

Launching the Server as an NT Service from a remote console

To start the Server as an NT service from a remote console, you must have administrative privileges on both machines and in the domains to which the machines belong, and PGPKeys must be installed on the console machine. In addition, the user name and password of the console user must be defined on both the local and remote machines.

For the Server's console to connect to the Server remotely, the **CommandPort**, **ServerSecureKeyID**, and **Allow KeyID command** parameters must be properly defined on the Server side. These parameters are required to establish security over a remote connection. If these parameters are not defined properly, you cannot check the configuration file or create the database on a remote Server. For more information, see the description for each of these parameters in [Chapter 2, "Configuration."](#)

1. In the Server's configuration file, specify a passphraseless key for **ServerSecureKeyID**. If this key is not passphraseless, the console cannot start a remote Server as an NT service because the NT service will prompt for a passphrase on the Server machine.

2. Identify the console's TLS key for **Allow KeyID command** in the Server's configuration file. This key must match the key selected as the console's signing key when the connecting console is invoked.
3. For security to work on the Server end of the connection, you must also define the private and public keyrings (**PrivateKeyRing** and **PublicKeyRing**) and Secure Mode (**SecureMode**) (for more information on Secure Mode, see the following section).
4. After setting up the security parameters, you can start the NT service remotely as follows:

To start the Server as an NT service on the remote machine, start the Server console (**Start -> Programs -> PGP Certificate Server -> PGP Certificate Server console**).

If you have multiple keys on your default keyring, you are prompted to select the console's TLS key. If you have only one key on that keyring, it is automatically selected.

If the selected key requires a passphrase, you are prompted to enter the passphrase.

5. Verify that the NT Service **Disabled** checkbox on the console's **Command** panel is not checked. The console cannot launch the Server as an NT service if this box is checked.
6. Specify the IP address or host name of the remote machine on which the Service is installed on the console's **Command** panel.
7. Verify that the **CommandPort** displayed on the console's **Command** panel is the **CommandPort** configured on the Server machine. If it is not, change the port to match the Server's configured port number.
8. Verify that the default **Configuration File** displayed on the console's **Control** panel is the file that you want to use. **Important:** The configuration file displayed on the console must reflect the location of the file on the Server machine. If it is not correct, enter the correct file. Note that the **Browse** button is disabled if the IP address specified in the **Command** panel is remote.
9. If this is the first time you are running the Server, click **Create Database** on the Server console's **Control** panel. The Server displays a dialog box when it has successfully created a database.

10. Click **Start** on the Server console's **Control** panel to start the Server as an NT service. The Server will prompt for a TLS key and passphrase if a passphraseless key has not been specified in the `ServerSecureKeyID` parameter in the configuration file. When starting the Server remotely, you will not be able to respond to the passphrase dialog. As a result, it is advisable to define `ServerSecureKeyID` in the configuration file with a passphraseless key. Note that the **Start** button becomes a **Stop** button.
11. The Server continues to run as an NT service even if the console is closed.
12. To stop the NT service, click **Stop**.

Secure Mode

The Server includes a Secure Mode that you can use to perform deletions and other administrative tasks. When the Server is in Secure Mode, the Server 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.

See [Chapter 6, "Secure Mode"](#) for instructions to set up and use Secure Mode.

Troubleshooting

If you experience problems, use the NT Event Viewer to help identify what is wrong (**Start -> Programs -> Administrative Tools (Common) -> Event Viewer**). Select **Log -> Application** from the menu bar. This will list events logged by each application. Each entry displayed in the viewer describes an event that happened on the local machine. Use the Event Viewer's help to learn about the information displayed in each column of the Viewer.

Configuring the Server

When you install the Server, the configuration settings are set to default values. There are a few settings that must be defined or the Server will not start (see [“Required parameters for the Server” on page 27](#) for details). There are also a few settings that you may want to modify before starting the Server. You define the values for these settings using the Web-based **Configuration/Monitoring wizard** or by editing the configuration file, *pgpcertd.cfg*, using your favorite text editor.

-
- ❑ **NOTE:** If you installed the Replication Engine, use the configuration file described in this section, *pgpcertd.cfg*, to configure the Replication Engine. (The setup program’s default setting includes the installation of the Replication Engine.)
-

Required parameters for the Server

The following parameters must be defined properly. If they are not, or if the files specified do not exist, the Server will not run:

PublicKeyRing — Defaults to “..\etc\PGPcertd-pubring.pkr”.

PrivateKeyRing — Defaults to “..\etc\PGPcertd-secring.skr”.

If these parameters are defined properly, the Server runs successfully and checks the configuration file for errors or omissions in other parameters.

The keyring files included with the product are provided to demonstrate how the product works. These keys should not be used in a production environment. You must modify the included keyrings to use your own, trusted keys. For more information, see [“Using the Configuration/Monitoring wizard” on page 31](#).

Note that to access the Server from a remote console, the Server must be configured with the public key of that remote console. For remote NT Service control, Windows NT requires that you be authenticated with Domain Administrative privileges for the domains of both machines, using the same username and password for both domains (if the connection spans two domains).

Configuring the Server's console

The console and Server can run on the same machine. If remote access to a Server is desired, PGPKeys must be installed on the console machine. PGPKeys defines a Default Key Ring location. The key or keys that the console uses to authenticate itself with various Servers should be on this keyring.

Whenever you invoke the console, the console uses the default settings in your registry (for example, **SecurePort**, **CommandPort**, etc).

When you start the console, you select the console's key pair for TLS authentication (the console uses the public key to authenticate its connection to the Server). If authentication fails, the Server closes the connection.

You also identify the hostname or IP address of the Server machine, the Server's command port, and the console's command timeout (this last value also identifies the **CommandTimeout** for the Server if it is started as an NT service from the console). If the Server is running on the local machine, use "localhost" (the string without quotes) or **127.0.0.1** for the hostname; when the Server and console are on the same machine, security is disabled on the command port to the local Server listening on that port.

A few other notes about the console:

- The console does not have a configuration file.
- You can change the Server parameters that are displayed on the console (for example, control port, secure port, and configuration file). When the console starts the Server, it sends any new parameter values to the Server; the new values override the Server's current settings and take effect the next time you start the Server. Use this feature to change the default settings for these parameters (this eliminates the need to edit the Server's registry).
- The configuration file is only user-definable if the Server is an NT service.

Required Server parameters for console connection

For the Server's console to connect to the Server remotely, the **CommandPort** parameter must be defined properly in the configuration file or on the command line, and the **ServerSecureKeyID** and **Allow Keyid command** parameters must be defined properly in the configuration file. These parameters are required to establish security over a remote connection. If these parameters are not defined properly, you cannot check the configuration file or create the database on a remote Server.

The command port for the Server defaults to port 4000. If another process uses this port, change the **CommandPort** setting to a different port in the configuration file.

For security to work on the Server end of the connection, you must also define the private and public keyrings (**PrivateKeyRing** and **PublicKeyRing**) and secure mode (**SecureMode**).

Note that local access to the Server is always insecure, so you do not need to define **ServerSecureKeyID** for local console access. However, you must always define the keyrings and command port.

Overriding settings in the configuration file

The Server's command line parameters override the settings in the configuration file, and the settings from the console override both the command line and configuration file settings.

Extracting Key IDs for configuration purposes

During the configuration process, you must identify the 32 or 64-bit key IDs for the **MustSigID**, **AllowSigID**, and **Allow keyID** configuration settings. 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 <asciarmor>
```

Table 2-1. Command Line Switches for the *pgpkeyid* Command

Switch	Description
-e	Lists the key ID of the encryption portion of the DSS/Diffie-Hellman key. If you do not use this switch, you receive the signing portion (DSS) of the key.
-k	Parses the key IDs from a PGP keyring.
-a	Parses the key IDs from an ASCII-armored key file.

The Server uses only signing keys for its configuration. Other PGP products have use for the encryption KeyID option.

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

```
pgpkeyid -k keyring.pgp > keyring.new
```

Verifying validity of new configuration settings

After you change values for the Server's configuration settings, use the Server's **Check Configuration** feature to verify that your changes are valid.

To verify that your changes are valid for an active Server or NT service:

- a. Start the Server's console (for details, see ["Starting the Server's console" on page 53](#)).
- b. Verify that the settings on the console's **Command** panel display values for the correct Server.
- c. Click **Check Configuration** on the console's **Control** panel. If there are any configuration errors, the errors are displayed on the **Events** panel.

NOTE: If you use the **Configuration/Monitoring wizard**, the validity check occurs whenever you use the **Save all changes** feature.

Setting up the Configuration/Monitoring wizard

The **Configuration/Monitoring wizard** (wizard) is a quick and easy-to-use interface used to configure the PGP Certificate Server. To use the wizard, a Web server must be running on the machine where the Server is installed.

The wizard is a CGI script that can be hosted by any Web server that supports the standard CGI protocols. For example, the wizard works with the following server products:

- Netscape's FastTrack Server 3.01
- O'Reilly Software's WebSite Professional 2.0
- Microsoft's Internet Information Server 2.0, 3.0, and 4.0 (IIS)
- Apache Server 1.3

The wizard is shipped in two formats:

- Windows executable file
- Perl 5.x script

In most cases you will use the executable file. However, in some cases you may need to use the raw Perl script.

During Server installation, if the installation program locates the Microsoft IIS, the installation program automatically adds support for that Web server, and the executable version of the wizard script is automatically set up for you.

To use a pre-existing Web server, you must modify the Web server's configuration file by adding the appropriate aliases or mappings for the Server's Web documents and CGI scripts.

To configure the alias for the Web documents (based on the default installation paths), enter the following lines in the Web server's configuration file or wherever your server configures aliases:

```
scriptalias /certserver/cgi-bin/ "c:/Program Files/Network Associates/PGPcertd/cgi-bin"
alias /certserver/docs/ "c:/Program Files/Network Associates/PGPcertd/Documentation"
alias /certserver "c:/Program Files/Network Associates/PGPcertd/web/htdocs"
```

-
- **NOTE:** For complete information on how to modify the Web server's configuration, consult the Web server's documentation.
-

The wizard also requires Perl, version 5.00307 or later. The installer will verify your version of Perl. If the installer suggests an upgrade, you can download a recent Perl from the following sites:

www.activestate.com

www.perl.com

Using the Configuration/Monitoring wizard

You can access the **Configuration/Monitoring wizard** by entering one of the following URLs in the location field of any Web browser:

`http://<hostname>[:<port>]/certserver/default.htm`

`http://<hostname>[:<port>]/certserver/`

Use the hostname and port given during the installation process. By default, the port is set to 8080.

Read the introductory information to learn how the wizard works, and then follow the on-screen prompts to progress through the various configuration settings. The configuration information that you supply through the wizard is stored in a configuration file (*pgpcertd.cfg*). If you prefer not to use the wizard or want to make some quick adjustments to a few configuration settings, you can use a text editor to edit this file. The wizard includes online help for all configuration settings.

The following section describes each configuration setting.

Examining and editing the configuration file

All of the configuration values are stored in a configuration file, *pgpcertd.cfg*. The file is normally stored in the following default location:

C:\Program Files\Network Associates\PGPcertd\etc\pgpcertd.cfg

-
- ❑ **NOTE:** If the Certificate Server is running as a Server rather than an NT service, the configuration file used is the last one specified via the console unless one is specified on the command line. If the Server is running as an NT service, the configuration file used is the last one specified on the console. In either case, if the console has never been used, the configuration file, *pgpcertd.cfg*, in the *etc/* subdirectory is used.
-
- ❑ **NOTE:** If your configuration file is corrupted or deleted, you can use the master configuration file, *pgpcertd-Master.cfg*, to restore the original settings.
-

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

- General configuration settings ([page 35](#))
- Database configuration settings ([page 38](#))
- Certificate policy configuration settings ([page 39](#))
- Replication Engine configuration settings ([page 41](#))
- Secure Mode configuration settings ([page 44](#))
- Access Configuration settings ([page 46](#))

You can edit the configuration file at any time. The changes take effect the next time you start the Server. To restart the Server after you edit the file, press the Restart button on the Server's console.

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.

Table 2-1. Configuration Settings

Setting	Purpose	Page #
AccessLogFile	Identifies the file where access statistics are logged.	page 47
AccessLogDetails	Controls the level of statistics recorded in the Access Log File.	page 47
Allow	Defines level of access for users.	page 47
AllowSigID	Identifies keys that are allowed when TrimSigs is turned on.	page 39
CacheEntries	Identifies the number of the database entries cached by the Server.	page 38
CommandPort	Console manages Server operations over this port.	page 35
CommandTimeout	Directs the Server to close the command port if an attempt to send or receive does not complete within this period.	page 36
CycleLogDay	Controls when the Access Log File is cycled (archived).	page 36
CycleLogTime	Controls the time of day that cycling of the Access Log File occurs.	page 36
CycleLogKeep	Controls the number of old Access Log Files that the Server retains.	page 36
DBCACHESize	Controls the database cache size in bytes.	page 38
DefaultAccess	Defines default access.	page 37
Directory	Identifies where the database files are located.	page 38
IdleSyncTimeout	Directs the Server to save the database cache to disk after the Server has remained idle for a specified number of seconds.	page 38
LogLevel	Controls the level of information recorded in the System Log File.	page 37
Mode	Identifies the file permissions associated with the database.	page 38

Table 2-1. Configuration Settings

Setting	Purpose	Page #
MustSigID	Identifies the signatures a key must have to pass the policy requirement.	page 39
PolicyFailures	Controls if rejected keys are sent to the pending bucket or returns an error.	page 40
Port	Identifies the port to listen to for regular LDAP connections.	page 37
PublicKeyRing	Identifies the file that contains the Server's Transport Layer Security (TLS) key and any keys specified by an Allow Keyid, MustSigID, or AllowSigID configuration value.	page 45
PrivateKeyRing	Identifies the PGP private keyring file that contains the private portion of the Server's TLS key.	page 44
RandSeedFile	Name of file to use to store persistent pseudo random seed.	page 45
ReadOnly	Controls read/write access to database entries.	page 38
RepdCommandPort	Identifies the port on the engine that the console uses to connect to the engine. The console manages the engine through this port.	page 42
RepdCommandTimeout	Directs the Replication Engine to close its command port if an attempt to send or receive on the command port does not complete within this period.	page 42
Replica	Identifies the location where the database contents are to be replicated.	page 43
ReplicationSecureKeyID	Identifies the key ID of the keypair to use as the key to authenticate the client side of all LDAPS connections and console side of the Replication Engine's command port connection (RepdCommandPort).	page 43
RepLogFile	Identifies the log file where changes are recorded for replication.	page 43
SecureMode	Controls if Secure Mode is required, optional, or disabled.	page 45
SecurePort	Identifies the port to listen to for TLS connections.	page 45
ServerSecureKeyID	Identifies the key ID of the keypair to use as the Server's LDAPS key and the console side of the Server's command port connection.	page 45

Table 2-1. Configuration Settings

Setting	Purpose	Page #
SizeLimit	Identifies the maximum number of matches returned for a query.	page 37
TimeLimit	Identifies the maximum number of seconds allocated for a query.	page 37
TrimPhotoIDs	Instructs the Server to remove PhotoIDs from submitted keys.	page 40
TrimSigs	Instructs the Server to remove unauthorized signatures from submitted keys.	page 40
TrimUsers	Instructs the Server to remove unsigned user IDs from submitted keys.	page 40

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 (“”).

General configuration settings

This section describes the general configuration settings for the following:

- Users that have access to the Server.
- How Server statistics are logged.
- Other settings that affect how the database responds to queries.

CommandPort <Port> — *Command Port*

Where <Port> is the port the Server console uses to connect to the Server to manage Server operations. Communication on this port is secure if connecting remotely. Security is disabled for access to a local Server.

Defaults to port 4000. If another process is using this port number, change this setting to a different port.

CommandTimeout <Seconds> — Command Timeout

Where <Seconds> is a number of seconds. If a send or receive over the CommandPort does not complete within this period, the Server closes the connection.

Defaults to 10 seconds.

CycleLogDay <frequency> — Day to Cycle Log

This setting controls when and if the Access Log File is cycled (archived). For more information about Access Log File cycling, see “Access Log File Cycling” on page 108.

- To cycle the Access Log File weekly, enter the day you want cycling to occur: **Monday, Mon, Tuesday, Tues, Wednesday, Wed, Thursday, Thurs, Friday, Fri, Saturday, Sat, Sunday, or Sun.**
- To cycle the Access Log File every day of the week, enter **daily**.
- To disable cycling, enter **never** (the Access Log File continues to grow in size).
- Defaults to “never”.

CycleLogKeep <number> — Logs to Keep

Use this setting to control the number of old Access Log Files that the Server retains. When the number of old logs in the Access Log File directory exceeds the value for <number>, the Server 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.

CycleLogTime <time> — Time to Cycle Log

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.

LogLevel <level> — Logging Level

Identifies the degree of information recorded in the Event Log File (note that this is not the same as the Access Log File). You can view the contents of the Event Log File to find out how the Server 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). [Table 2-2](#) describes valid values for the LogLevel setting.:

Table 2-2. Values for LogLevel

Value	Description
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 Event Log File, each entry generated by the Server has a source of "PGPCERTD." This distinguishes these messages from those generated by other processes.

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.

SizeLimit <size> — Size Limit

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

TimeLimit <seconds> — Time Limit

Identifies the maximum number of seconds, in real time, that the Server 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).

Database Configuration Settings

CacheEntries <*number of entries*> — *Cache Entries*

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

DBCacheSize <*size*> — *DB Cache 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.

Directory <*path*> — *Database Directory*

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.

Defaults to “..\data”.

IdleSyncTimeout <*seconds*> — *Idle Sync Timeout*

Identifies the number of seconds the Server 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.

Mode <*file permissions*> — *Database File Permissions*

This setting, which identifies the file permissions associated with newly created database files, is always 0600 on NT Certificate Servers. The default setting, 0600, gives read and write access by the file's owner.

ReadOnly on | off — *Database Access Mode*

Controls if clients can read and write entries to the database, or if they are restricted to read-only access. This setting is useful when replicating data to multiple Servers, and you want to grant the ability to search for and retrieve entries, but prevent users from adding or modifying entries. 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. By default, the read-only setting is turned off, which means clients have read and write access to the Server.

Certificate 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 Server will accept the key, and which signatures are allowed to remain on a submitted key. For information on gathering key IDs, see “Extracting Key IDs for configuration purposes” on page 29.

AllowSigID <keyID> — Allowed Certificate Signatures

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 Server. You can place multiple AllowSigID lines in the configuration file and each are treated with equal significance.

-
- ❑ **NOTE:** Before you start the Server, make sure all of the required certificates are stored on the Server or in the Server’s public keyring (see “PublicKeyRing <filename> — Public Key Ring” on page 44).
-

MustSigID <keyID> — Required Certificate Signatures

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

-
- ❑ **NOTE:** Before you start the Server, make sure all of the required certificates are stored on the Server or in the Server’s public keyring (see “PublicKeyRing <filename> — Public Key Ring” on page 44).
-

PolicyFailures pending | error — *Action on Key Policy Failure*

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 “pending,” the key is stored in the pending bucket. If set to “error,” the key is ignored and an error message is generated. The “error” setting is useful for sites that do not want to maintain a pending bucket. The default setting is “pending.”

TrimPhotoIDs yes | no — *Remove PhotoIDs*

Allows you to remove a PhotoID from a key before the key is stored on the Server. 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 Server. The default setting is “no.”

TrimSigs yes | no — *Remove Unallowed Signatures*

Allows you to remove unauthorized signatures from the UserIDs on a key before it is stored on the Server. 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.
-

TrimUsers yes | no — *Remove Unallowed User IDs*

Allows you to remove unauthorized user IDs from a key before it is stored on the Server. When this setting is turned on (that is, set to yes), only user IDs that still have a signature (not counting the self-signature) are kept on the key. All other user IDs are trimmed. The default setting is “no.”

Certificate policy configuration matrix

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

Table 2-3. Certificate Policy Configuration Matrix

MustSigID	AllowSigID	TrimUserID	TrimSigs	Server Results
Not set	Any or no value	No	No	The Server accepts all keys regardless of how they are signed, and performs no trimming.
Set	Any or no value	No	No	The Server 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 Server 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 Server 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 Engine configuration settings

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

The following parameters are required for secure access from the Replication Engine's console to the Replication Engine over a remote connection: **SecureMode**, **PublicKeyRing**, **PrivateKeyRing**, **RepdCommandPort**, **Allow keyid** command, and **ReplicationSecureKeyID**. If these values are not defined, only insecure local access to the Replication Engine is allowed. If **RepdCommandPort** is not defined, you cannot access the console.

-
- **NOTE:** For the balance of this guide, the terms Replication Engine and engine are used interchangeably.
-
- **NOTE:** When you use the engine, you identify the Servers 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.
-

The following are the relevant configuration settings for replication.

Allow keyid <KeyID> command

Where <KeyID> is the 32 or 64 bit KeyID of a PGP key.

Use Allow keyid to identify the consoles (identified by their TLS keys) that are allowed access to the Server or engine through the command port.

RepdCommandPort <command port> — *Replication command port*

Where <Port> is the port the Replication Engine's console uses to connect to the engine to manage engine operations. Communication on this port is secure if connecting remotely. Security is disabled for local access to the engine.

Defaults to port 5000. If another process uses this port number, change this setting to a different port.

RepdCommandTimeout <seconds> — *Replication command timeout*

Where <Seconds> is a number of seconds. If a send or receive to the console does not complete within this period, the engine closes the connection.

Defaults to 10 seconds.

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

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 Server must be running an MIT style public key server. When replicating to more than one Server, list multiple Servers on the same line, or create a separate entry in the configuration file for each Server. If you do not specify a port number, the default port for that protocol is used. The protocol defaults to ldap.

ReplicationSecureKeyID **<KeyID>** — *Replication Secure Key ID*

Specifies the key ID of the keypair to use as the key to authenticate the client side of all LDAPS connections. 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.

This also specifies the key to use to authenticate with the console over the command port. If not specified, only insecure local access to the engine is allowed.

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

To connect a console to a remote engine running as an NT service, this key must be passphraseless.

If this key is not defined and you run the engine from the command line, the software prompts you for a key.

RepLogFile **<filename>** — *Replication Log File*

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

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

For details on how to export data from one Certificate Server to another, see [Chapter 3, "Operation."](#)

- ❑ **NOTE:** If a filename includes one or more spaces, you must enclose the entire name in quotes.
-

Secure mode configuration settings

Use the following configuration settings to operate the Server in *Secure Mode*. Use Secure Mode to perform administrative functions such as the deletion of keys. When the Server is in Secure Mode (that is, the value for the SecureMode setting in the configuration file is Required), the Server 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 [“Setting Up Secure Mode” on page 93](#).

PrivateKeyRing <filename> —Private Key Ring

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 Server was started. Use PGP to create this file. To enable support for TLS, the private portion of the Server’s TLS key must be in this keyring.

Defaults to “..\etc\PGPcertd-secring.skr”.

PublicKeyRing <filename> — Public Key Ring

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 Server was started.

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

When the Server is in Secure Mode (that is, the value for the SecureMode setting in the configuration file is Required), the Server 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, **ServerSecureKey**), **SecureMode** and the **SecurePort** are disabled and the console can only connect via local access.

Defaults to “..\etc\PGPcertd-pubring.pkr”.

For find out how to set up Secure Mode, see “Setting Up Secure Mode” on page 93.

RandSeedFile <filename> — *Random seed file*

Name of file to use to store persistent pseudo random seed. Defaults to Windows NT directory (for example, C:\WinNT\randseed.rnd).

SecureMode <Mode> — *Secure Mode*

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

- Disabled — Turns off Secure Mode.
- Required — The Server cannot start unless it can successfully provide secure access (Transport Layer Security (TLS)). When this setting is used, there must be a Server key in the **PublicKeyRing** and **PrivateKeyRing** keyring files. In addition, the operator must enter the passphrase for the secret Server key each time the Server starts or the Server key must be passphraseless.
- Optional — If the Server is started in auto-start mode (-s command line argument is present or the Server is started as an NT service), then the Server enables TLS if TLS can be enabled without user intervention (that is, no passphrase is required). Otherwise, the Server starts with TLS disabled. If the Server 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.

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).

SecurePort <Port> — *Secure 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).

ServerSecureKeyID <KeyID> — *Server Secure Key ID*

Identifies the key ID of the keypair to use as the Server’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 first public/private keypair found in the keyring is used.

This is also the key the Server uses to authenticate with a console connecting through the command port. If not specified, only insecure local access to the Server is allowed.

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

Use this parameter to identify the KeyID of the Server's TLS key. If you do not define this parameter, the Server asks you to select a key when the Server starts.

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

If the Server is running as an NT service, Secure Mode is optional, and the key requires a passphrase, the Server disables the secure port and allows only insecure console connections from the local host over the command port.

If the Server is running as an NT service, Secure Mode is required or the key requires a passphrase, and the NT service is not configured to interact with the desktop, the Server exits.

If the Server is configured to interact with the desktop or the Server is invoked as a command line application, Secure Mode is required, and the key has a passphrase, the Server prompts for the passphrase.

IMPORTANT: To have a console start a remote Server running as an NT service, the Server's key must be passphraseless.

Access configuration settings

-
- ❑ **NOTE:** When you establish access controls, there are two levels of access of concern. First, you use the **Allow** configuration setting to define the type of access a user or group of users has to various Server functions. Second, you use the **MustSigID** configuration setting to restrict the keys that can be stored on the Server by requiring them to be signed by a given key ID.
-

AccessLogDetails <type> — Items to Log in Access Log

Controls the type of Server operations recorded in the Access Log File. You must list each operation that you want recorded in the Access Log File, and you must separate operations with a space. [Table 2-4](#) describes valid values for AccessLogDetails.

Table 2-4. Valid Values for AccessLogDetails

Value	Description
none	No information is recorded in the access log.
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 Certificate Server.
all	Record all of the operations listed above.

AccessLogFile <filename> — Access Log File

Identifies the relative path or absolute pathname for the Access Log File. By default, there is no Access Log File. If you want one, use this setting to name the file.

Allow keyid <KeyID> command

Where <KeyID> is the 32 or 64 bit KeyID of a PGP key.

Use Allow keyid to identify the consoles (identified by their TLS keys) that are allowed access to the Server or engine through the command port.

Allow <who> <access> — Allow access by

Identifies users that have a specific kind of access to the Server.

<who> Parameter — *This entity is*

The <who> parameter identifies a user or group of users with a specific IP address, hostname, or keyID (32 or 64-bit):

Allow ip <IP Address> <access>

Allow host <Hostname> <access>

Allow keyid <KeyID> <limited access level>

Where

- <IP Address> is the dotted decimal IP address (for example, 127.0.0.1).
- <Hostname> is the Server's TCP/IP hostname (for example, certserver.pgp.com).
- <KeyID> is the 32 or 64 bit KeyID of a PGP key. When a keyid is specified for the <who> parameter, only the add and delete and command access settings are valid.

The first parameter (ip, host, and keyid) identifies the method used to identify the user, and the second parameter is the IP address, hostname, or key IDs. For example, to identify a user or group of users by their IP address, enter "ip" followed by the appropriate IP address.

Use wildcard characters to include a range of users that fit a given criteria.

Use the **Allow keyid <keyid> command** to identify the console's TLS keys that are allowed access to the Server.

<access> Parameter — *Access Permitted*

The <access> parameter identifies the level of access granted to the users identified in the following manner:

1. By an ip or host <who> parameter, where <access> is none, compare, search, read, add, delete, or all.
2. By the keyid <who> parameter, where <access> is Delete, Command, or GroupUpdate.

An <access> of Delete means that strongly-authenticated requests from the specified KeyID will be allowed to delete or disable keys on the Server. The special value "self" can be used in place of an actual KeyID to allow a key's owner to delete or disable his own key.

An <access> of Command means that a console that authenticates with a specified KeyID will be allowed access to the Server via the command port.

To specify the console that is allowed access to the Server on the command port, use the following syntax:

Allow keyid 0x9615A02DBBE1E0E2 command

An `<access>` of GroupUpdate allows an administrator running PGPKeys to send the administrator's list of groups to the Server to replace the list of groups that is already there. The administrator must be the owner of the key specified by the indicated keyid.

Table 2-5 describes the levels of access, which are hierarchically accumulative (that is, each level of access automatically includes all of the permissions granted by the lower levels of access in the hierarchy).

Table 2-5. Descriptions of Access Levels

Access Level	Description
none	Denies all access to the specified user.
compare	If the value is known, it can be compared against the value in the database.
search	Allows the designated users to search the contents of the database if searching from an LDAP client (searching from a PGP client requires read access).
read	Allows the specified user to query and retrieve keys from the Server. The following example gives read access to all users: allow ip * read
add	Allows the specified user to query and retrieve keys and to add new keys to the Server. The following example gives read access and add access to all users who reside at pgp.com: allow host *.pgp.com add
delete	Allows the user to retrieve, add, and delete keys from the Server. Users with "delete" permission can delete keys from the Server if they are using a key with a signature authorized to perform this operation. The following example gives read, add, and signed deletes to the users at the designated address: allow ip 205.180.136.115 delete Although users with "delete" permission can perform signed deletions, they are not authorized to perform LDAP deletes. See note below.

Table 2-5. Descriptions of Access Levels

Access Level	Description
all	Allows the specified users to perform all of the above functions. They can also use the standard LDAP functions (add, delete and modify) to manipulate data stored in the database. This setting is not normally used with the Server, but is provided for those sites that intend to build their own LDAP front-end to access the Server's directory.

❑ **NOTE:** Delete authority requires two configuration changes. You must allow the host or IP to perform deletes (use an “allow host” or “allow ip” line), and you must indicate what PGP key must sign the delete request (use an allow KeyID line).

❑ **NOTE:** 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 (*.pgp.com).

DefaultAccess none | compare | search | read | add | delete | all — Default Access

Identifies the default level of access granted to all users who are not covered by the access permissions specified with the “Allow” setting. [Table 2-6](#) describes valid values for DefaultAccess:

Table 2-6. Valid Values for DefaultAccess

Value	Description
none	Denies all access to default users.
compare	If the value is known, it can be compared against the value in the database.
search	Allows default users to search the contents of the database.
read	Allows default users to query and retrieve keys from the Server.
add	Allows default users to query and retrieve keys and to add new keys to the Server.

Table 2-6. Valid Values for DefaultAccess

Value	Description
delete	Allows default users to retrieve, add, and delete keys from the Server. Users with “delete” permission can delete keys from the Server if they are using a key with a signature authorized to perform this operation. Although users with “delete” permission can perform signed deletions, they are not authorized to perform LDAP deletes.
all	Allows default users to perform all of the above functions. They can also use the standard LDAP functions (add, delete, and modify) to manipulate data stored in the database.

This chapter describes how to start and stop the Server, the Server's console, the Replication Engine, and the Replication Engine's console.

Starting the Server's console

-
- ❑ **NOTE:** To start the Server as an NT service you must have NT administrative access privileges on the machine. Domain administrative privileges are required to start and stop the Server remotely.
-
- ❑ **NOTE:** **PGPKeys** must be installed on the machine running the Server's console. If **PGPKeys** is not installed, only insecure local access to the Server is allowed. **PGPKeys** defines a Default Keyring location. This keyring contains the key or keys with which the Server's console can authenticate itself with various Servers. **PGPKeys** is a component of the PGP Desktop Security product.
-

You can start the Server's console from the command line or from the **Start** menu. The console and Server must be compatible versions. If the console attempts to connect to a Server that is not compatible, an error occurs.

Note that you can run multiple consoles on the same machine. The first console to make a connection to a given Server gains control of the Server.

TLS secures the command port and allows only authorized users access to a remote Server. Use the **Allow keyid command** to modify the Server configuration file and specify the console TLS key(s) that are allowed access to the Server through the command port (see "[Allow <who> <access> — Allow access by](#)" on page 2-47).

Starting the Server's console from the Start menu

To start the PGP Certificate Server Console from the Start menu:

1. Choose **Start -> Programs -> PGP Certificate Server -> PGP Certificate Server Console**. The Server's console displays the **Control** panel and other tabs.

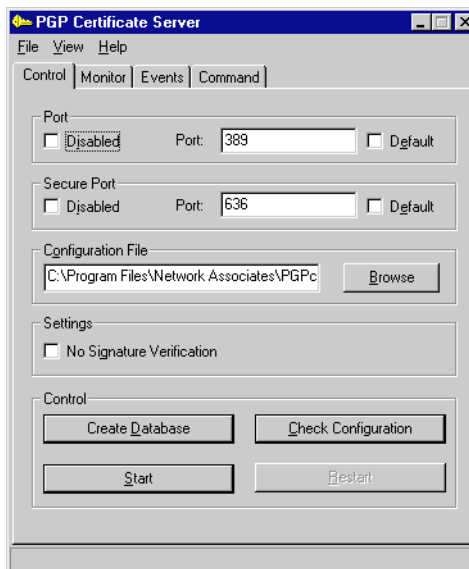


Figure 3-1. Certificate Server Console - Control Panel

Starting the Server's console from the command line

The Console takes the following command line parameters:

```
[ -a ] [ -b <CommandPort> ] [ -i <IPAddress> ] [ -g ] [ -k <CommandTimeout> ] [ -t <secure port> ] [ -p <control port> ]
```

Option	Description
-g	Disables the NT Service (checks the "NT Service Disabled" checkbox in the Command tab of the console). It is not checked by default.
-a	Tells the Server not to verify signatures (checks the "No Signature Verification" checkbox in the console). It is off by default
-t <port>	Identifies the port number that the Server listens to for Secure Mode. If this parameter is not specified on the command line, the Server uses default values or values in the registry. Specifying -1 disables Secure Mode. A port number of 636 enables the default LDAPS port.

Option	Description
-p <port>	Identifies the port number that the Server listens to for client requests and certificate submittals. If this parameter is not specified on the command line, the server uses default values or values in the registry. Specifying -1 disables the regular port. A port number of 389 enables the default LDAP port.
-b <CommandPort>	Defines the Command Port that the console attempts to connect to to send commands to the Server. This should be the same as the Server's Command Port.
-i <IPAddress>	Defines the IP address or hostname of the machine on which PGPcertd is installed.
-k <CommandTimeout>	Defines the timeout over the command port on the console side. If the console attempts to transmit or receive over the command port, it closes the connection if this timeout expires before the transmit or receive completes. The default value for CommandTimeout is 10 seconds. When you start an NT service from the console, this is also the timeout for the service's command port.

Starting the Certificate Server

There are two ways to run the Certificate Server: as a command line application and as an NT service.

Command line application - You can run the Server from the NT command line (see [“Starting the Server’s console from the command line”](#) on page 54).

NT service - The installation program installs the Server as an auto-starting NT service but does not start it. The NT service starts automatically at boot time. This feature allows the Server to continue to run when you log out of your NT account. Note that if there are errors in the Server's registry settings or configuration file, the NT service does not start.

You can also start the NT service from the console if the NT Service **Disabled** checkbox on the **Command** tab of the console is not checked.

Note the following:

- If the Server is running as an NT service, **SecureMode** is Optional, and the Server's key requires a passphrase, the Server disables the secure port and allows only insecure console connections from the local host over the command port (an NT service cannot bring up a GUI to obtain the passphrase).

- If the Server is running as an NT service, **SecureMode** is Required, and the Server's key requires a passphrase, a passphrase dialog is displayed and the NT service does not start until the passphrase is entered.

Starting the Certificate Server as an NT service from the console

You can start the Server as an NT service from the console.

1. Verify that the configuration file name displayed on the console's **Control** panel is correct. To do so, display the Server's console (**Start -> Programs -> PGP Certificate Server -> PGP Certificate Server console**). The console's **Control** panel displays the Server's configuration file. If required, correct the entry. Note that the path must be the path on the machine where the NT service resides.
2. Verify that the NT Service **Disabled** checkbox on the **Command** panel is not checked.
3. Verify that the other information displayed on the **Command** panel is correct. Make any required corrections.
4. If this is the first time you are starting the Server, you must create the database. Click **Create Database** on the **Control** tab.
5. Click **Start** on the **Control** panel. Note that the **Start** button becomes a **Stop** button.

When the console starts the Server, it sends the Server several parameter values that are settable on the console. These parameters override the Server's command line and configuration file settings.

If you are accessing the Server remotely, the console connects to the Server's command port, the Server extracts the keyid of the console's public key and compares it against a list of keyids authorized to connect to the Server (that is, the **Allow KeyID command** setting in the configuration file). If the console's keyid is not on this list, the Server rejects the connection to the console and returns to listening for new console connections over the command port.

Stopping the NT service

To stop the NT service:

1. Verify that the NT Service **Disabled** checkbox on the console's **Command** panel is not checked.
2. Click **Stop** on the console's **Control** panel.

Starting the Server from the command line

You can also start the Server from the NT command line. When you start the Server in this manner, you specify the Server's run-time parameters. You must use the **-g** command line switch. You must also use the auto-start option, **-s**, to tell the Server to start accepting connections. If you do not use the **-s** option, you must connect to the Server using the console and press the **Start** button.

Note that Servers invoked from the command line can be controlled from the console.

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>] [-b <port>] [-k <command timeout>]
```

-
- NOTE:** The first time you start the Server you must create the database. As a result, you must enter the following: **pgpcertd -g -n**.
-
- NOTE:** The **-g** option is required to start the software as a Server rather than an NT service.
-

The following table describes each of the command line switches.

Table 3-1. Server Command Line Switches

Command Line Switch	Description
-a	Instructs the Server 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 Server to another.
-c	Checks the current configuration file for accuracy. Does not start the Server. When you make configuration changes, use this option to verify that the new configuration values are valid.
-n	Causes the Server to consult the “directory” setting in the configuration file to determine where to create new database files. You must use this switch the first time you run the Server and when you change the location of the database file. If you do not, an error occurs.
-s	Automatically starts the Server after the console loads. Note that if this option is used and there are errors in the configuration file, the Server ignores this option. For more information, see “Required parameters for the Server” on page 27 .
-t <port>	Identifies the port number that the Server listens to for Secure Mode. Defaults to port 636. If -t is not present, the Server uses the value for SecurePort found in the configuration file. -t -1 (minus one) disables LDAPS and Secure Mode.
-f <file>	Identifies the configuration file that the Server uses. If -f is not specified, uses default.
-p <port>	Identifies the port number that the Server listens to for client requests and certificate submittals. Defaults to port 389.

Table 3-1. Server Command Line Switches

Command Line Switch	Description																														
-d <i><level></i>	<p>Turns on the debug mode and provides a level of information based on the level you select. The following are the debug levels:</p> <table border="1" data-bbox="489 357 821 944"> <thead> <tr> <th>Debug Level</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Trace</td> </tr> <tr> <td>2</td> <td>Packets</td> </tr> <tr> <td>4</td> <td>Arguments</td> </tr> <tr> <td>8</td> <td>Connections</td> </tr> <tr> <td>16</td> <td>Data encodings</td> </tr> <tr> <td>32</td> <td>Search filters</td> </tr> <tr> <td>64</td> <td>Configuration</td> </tr> <tr> <td>128</td> <td>Access</td> </tr> <tr> <td>256</td> <td>Statistics</td> </tr> <tr> <td>512</td> <td>Statistics (more)</td> </tr> <tr> <td>1024</td> <td>Shell</td> </tr> <tr> <td>2048</td> <td>Parsing</td> </tr> <tr> <td>8192</td> <td>PGP Errors</td> </tr> <tr> <td>65535</td> <td>All</td> </tr> </tbody> </table> <p>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.</p>	Debug Level	Description	1	Trace	2	Packets	4	Arguments	8	Connections	16	Data encodings	32	Search filters	64	Configuration	128	Access	256	Statistics	512	Statistics (more)	1024	Shell	2048	Parsing	8192	PGP Errors	65535	All
Debug Level	Description																														
1	Trace																														
2	Packets																														
4	Arguments																														
8	Connections																														
16	Data encodings																														
32	Search filters																														
64	Configuration																														
128	Access																														
256	Statistics																														
512	Statistics (more)																														
1024	Shell																														
2048	Parsing																														
8192	PGP Errors																														
65535	All																														
-b <i><command port></i>	<p>Identifies the port number (Command Port) that the Server listens to for console connections and commands. Command Port must be set in the registry or on the command line or configuration file. Defaults to 4000 or the last value that was used successfully. Multiple Servers running on the same machine must specify different Command Port numbers.</p>																														
-k <i><command timeout></i>	<p>Identifies the amount of time, in seconds, that the Server waits for sends and receives to complete over the Command Port. The Server closes the connection if the CommandTimeout expires while it is sending or receiving data over the Command Port. The command timeout defaults to 10 seconds.</p>																														
-g	<p>Tells the software to start as a command line Server rather than an NT service. Required if the Server is invoked on the command line.</p>																														

Configuring the Server to start automatically

If you do not want to manually start the Server from the console, you can configure the Server to start automatically. You can do this by invoking the Server from the command line with the `-s -g` options.

An NT service that comes up at boot time has no command line. As a result, the NT service automatically starts the Server when the NT service starts.

The NT service will fail to start the Server if it encounters a configuration problem. If this happens, you must resolve the problem and manually start the Server from the console, from the Services applet in the Control Panel, or by rebooting.

Invoking the Server with the `-s` auto-start option

Note that if the Server is invoked from the command line with the auto-start option (`-s`) and there are errors in the configuration file, the Server ignores the auto-start option. (The `-s` option automatically starts the Server after the Server initializes. For details, see [“Starting the Server from the command line” on page 57.](#))

Verifying that the Server is running

To monitor the activities that are taking place on the Server, click the console's **Monitor** tab. To check on the Server's activities and to see if any errors have occurred, click the **Events** tab.

For information on configuring and examining the **NT Event Log** file, see [Chapter 7, “Monitoring and Logging.”](#)

Running multiple Servers on the same machine

You can start any number of Servers from the command line as long as each Server uses a different set of port numbers database directories, access log files, public keyring files, private keyring files, and replication log files. You can only run one Server as an NT service.

Starting the Replication Engine's console

-
- **NOTE:** To run the PGP Replication Engine, you must have NT administrative privileges. Domain administrative privileges are required to start or stop the engine remotely as an NT service.
-

- **NOTE: PGPKeys** must be installed on the machine running the engine's console. If **PGPKeys** is not installed, only insecure local access to the engine is allowed. **PGPKeys** defines a Default Keyring location. This keyring contains the key or keys with which the engine's console can authenticate itself with various engines. **PGPKeys** is a component of the PGP Desktop Security product.
-

The Replication Engine's console allows you to start the Replication Engine and specify options that control how the Replication Engine functions. The Replication Engine's console can also be started from the command line. The engine can be connected to one console at a time.

To start the PGP Replication Engine console:

1. Verify that **PGPKeys** is installed on the machine running the engine's console.
2. Choose **Start -> Programs -> PGP Certificate Server -> PGP Replication Engine console**.

The PGP Replication Engine console displays the **Control** panel. Use the **Control** panel to start and stop the Replication Engine and to set initialization options. Use the tabs at the top of the console to access the **Monitor** panel (shows the activities taking place on the Replication Engine), **Events** panel (shows how well the Replication Engine is working), and **Command** panel (shows NT Service and Replication Engine information).

-
- **NOTE:** Complete details regarding the **Command** panel appear in [Chapter 5, "Using the Replication Engine."](#) Complete details for the **Monitor** and **Events** panels appear in [Chapter 7, "Monitoring and Logging."](#)
-

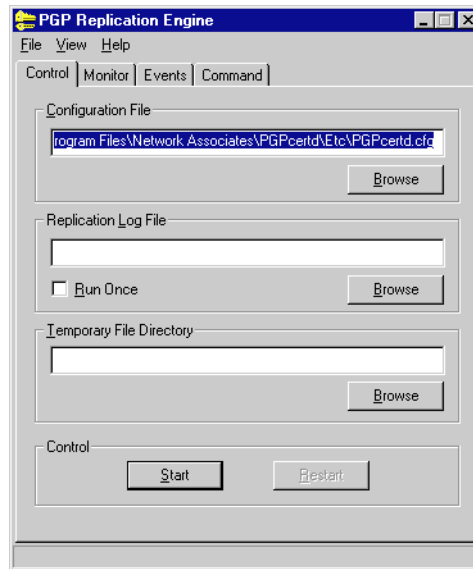


Figure 3-2. Replication Engine Console - Control Panel

Preparing to start the Replication Engine

Before you start the Replication Engine, you can identify the following directory and files:

- The configuration file that you want the Replication Engine to use
- The replication log file (this file contains a record of all changes to the database on the master Server)
- The directory where the Replication Engine's temporary files are stored
- The Replica parameter must be defined in the configuration file (for more information, see [“Replica \[<protocol>://<hostname> or <IPaddress>\[:<port>\] — Hosts to Replicate Database to”](#) on page 2-43).

The first time you run the console, these settings are set to their default values. The configuration file, replication log file, and temporary directory are user-definable if the engine is running as an NT service. If the engine is not running as an NT service, you cannot define the configuration file via the console. Note that your entries must reflect the pathnames on the Server machine. The **Browse** buttons are disabled if the specified Server is running on a remote machine. The following table describes these settings.

Table 3-1. Directory and Files used by Replication Engine

Directory or File	Description
Configuration File	Identifies the configuration file you want the Replication Engine to use. By default, the Replication Engine uses the Certificate Server's configuration file, <i>pgpcertd.cfg</i> . This file contains all of the configuration settings that affect the Replication Engine. Note that if the NT Service Disabled checkbox is checked (console's Command panel), you cannot change the configuration filename.
Replication Log File	Identifies the replication log file you want the Replication Engine to use (this file contains a record of all changes to the database on the master Server). This option overrides a value in the configuration file. During normal operation, the Replication Engine continuously monitors the replication log file for more entries. When you select the Run Once option, the Replication Engine looks at the replication log file once only and does not wait for more replication entries to be added.
Temporary File Directory	Identifies the directory you want the Replication Engine to use as a temporary file directory. By default, the Replication Engine uses the values in the TEMP or TMP environmental variables to identify the temporary file directory. If these variables are not defined, the files are stored in the current directory.

-
- ❑ **NOTE:** If you shut down the Replication Engine (*pgprepd*) and, when you restart the Replication Engine, you do not want to continue to process the database entries where you left off, remove all of the *pgprepd* related files from the temporary directory.
-

- ❑ **NOTE:** 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 **Temporary File Directory** option (or **-t** in the command line) to explicitly designate where these temporary files are stored.
-

Starting the Replication Engine from the console

To start the engine from the engine's console, click **Start** on the **Control** panel. Notice that the **Start** button turns into a **Stop** button.

If the engine is running as an NT service and the configuration file and temporary file directory are not identified on the console's **Control** panel, the engine uses the configuration file and temporary file directory settings found in the registry. If the engine's log file is not identified on the console's **Control** panel, the engine uses the setting found in the configuration file.

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: start the engine from the command line and select the key when prompted, or define the `ReplicationSecureKeyID` parameter in the configuration file.

- 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 requires a passphrase, the engine is run as an NT service, and the NT service is configured to interact with the desktop, the engine prompts for the key's passphrase.
- If the selected key does not require a passphrase, the engine or NT services does not prompt for one.
- If the engine does not receive a required passphrase because you cancel the passphrase dialog box, the engine allows only insecure console connections from the local machine over the command port.

Starting the Replication Engine from the command line

You can also start the Replication Engine from the NT command line. You must use the `-g` command line switch. You must also use the auto-start option, `-s`, to tell the engine to start accepting connections. If you do not use the `-s` option, you must connect to the engine using the console and press the **Start** button.

When you use this method, the engine runs with the settings you enter on the command line, and the values displayed on the engine's console reflect those values:

```
pgprepd -g [-f <file>] [-t <directory>] [-r <file>] [-c] [-o] [-d<level>] [-s] [-b  
<command port>] [-k <command timeout>]
```


Table 3-2 describes the Replication Engine's command line switches.

- **NOTE:** The **-g** option is required to start the software as an engine rather than an NT service.
- **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 [Table 3-2](#).

Table 3-2. Replication Engine Command Line Switches

Command Line Switches	Description
-f <file>	Identifies the configuration file that you want the Replication Engine to use. By default, the Replication Engine uses the Server's configuration file, <i>pgpcertd.cfg</i> . 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.
-r <file>	Identifies the log file you want the Replication Engine to use. This option overrides the value in the configuration file.
-c	Checks the current configuration file for accuracy. Does not start the Replication Engine. When you make configuration changes, use this option to verify that the new configuration values are valid.
-o	During normal operation, the Replication Engine continuously monitors the replication log file for new entries. When you use the -o option, the Replication Engine looks at the replication log file once only. When you use this option, you must also use the -r option.
-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 Certificate Server, <i>pgpcertd</i> .
-s	Automatically starts the Engine when the engine is invoked.
-b <CommandPort>	Defines the engine's Command Port over which the console attempts to connect to to send commands to the engine.
-k <CommandTimeout>	Defines the timeout over the command port. If the engine attempts to transmit or receive over the command port, it closes the connection if this timeout expires before the transmit or receive completes. The default value for CommandTimeout is 10 seconds.

Table 3-2. Replication Engine Command Line Switches

Command Line Switches	Description
-g	Required if the engine is invoked from the command line. Indicates that the engine is not running as an NT service.

Verifying that the Replication Engine is running

When you start the Replication Engine, the **Start** button on the console's **Control** tab changes to a **Stop** button. At this point, you can advance to the **Monitor** panel to see the activities the Replication Engine is processing. You can also advance to the **Events** panel to see the Replication Engine's activity.

To find out how to configure and examine the **NT Event Log** file, see [Chapter 7, "Monitoring and Logging."](#)

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 port numbers database directories, access log files, public keyring files, private keyring files, and replication log files. You can only run one engine as an NT service.

Stopping the Replication Engine from the console

To stop the Replication Engine, click the **Stop** button on the console's **Control** tab.

Learning about Servers

Server and console compatibility

A Server can be connected to only one console at a time. The Server detects a console's version number when the Server and console connect over the command port. If the console version number is incompatible with the Server, the Server rejects the connection.

Server, PGPkeys, and the console

A Server can be connected to both a **PGPkeys** client and a console at the same time, as long as each uses a different port to the Server. When **PGPkeys** executes a Server operation (for example, an add, search, or delete), the console updates its monitor and event log displays to reflect the results of the operation.

Console retrieves Monitor and Event data

The console retrieves Monitor data only if the console is connected to the Server and the Server is running. Event Log data is retrieved when the console receives error results from **Check Configuration**, **Create Database**, or **Start** operations, or when the Events tab is selected while the console and Server are connected.

Server times out and closes connection

When the console and Server are connected over the command port, the Server closes the connection if the connection is inactive for a span of thirty minutes (that is, the console does not send a command to the Server during that time period). This feature gives other consoles a chance to connect to the Server.

Note that if **Auto Update** in the console's **Monitor** panel is set to request updates frequently and the **Monitor** panel is active, the Server does not time out and close the connection. In this situation, the console can monopolize the Server's command port, preventing another console from connecting with the Server.

Server authentication

The Server's TLS key is selected from the keys in the Server's keyring. The location of the keyring is defined in the configuration file. There are two ways to select the Server's TLS key: start the Server from the command line and select the key when prompted, or define the `ServerSecureKeyID` parameter in the configuration file.

- If the selected key requires a passphrase and the Server is run as a command line application, the Server prompts for a passphrase.
- If the selected key requires a passphrase, the Server is run as an NT service, and the NT service is configured to interact with the desktop, the Server prompts for the key's passphrase.
- If the selected key does not require a passphrase, the Server or NT services does not prompt for one.
- If the Server does not receive a required passphrase because you cancel the passphrase dialog box, the Server allows only insecure console connections from the local machine over the command port.

Configuring the Server to start automatically

If you do not want to manually start the Server from the console, you can configure the server to start automatically. You do this by invoking the Server from the command line with the `-s` option.

An NT service that comes up at boot time has no command line. As a result, the NT service automatically starts the Server when the NT service starts.

The NT service will fail to start the Server if it encounters a configuration problem. If this happens, you must resolve the problem and manually start the Server from the console or reboot.

HTTP support for PGP 5.0 clients

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

Your IIS4 Web server will be pre-configured as an HTTP gateway to support the PGP client. The permissions are configured to deny public access to these interfaces for security reasons. Use Windows Explorer and the Properties panel to grant filesystem access for the Web server to the following files, relative to the directory where you installed PGP Certificate Server.

Web\CGI-BIN\Add.exe

Web\CGI-BIN\Lookup.exe

To set up the HTTP gateway, you must take the following steps:

- Select the machine where the Web server will reside
- Select the port number the HTTP gateway will listen to
- Select an existing or dedicated Web server

PGP Certificate Server will serve requests from this gateway as if the requests originated from the Web server machine. If you want to enable the HTTP gateway, make sure you understand the security implications and configure your network security policy and PGP Certificate Server policy accordingly.

MIT-style key servers act as limited Web servers. By default, they listen to port 11371 rather than the standard port of 80. For PGP 5.0 clients to work with the default port, the HTTP gateway must be set up to run on a Web server listening to port 11371. However, a simpler method to make this work is to set up the HTTP gateway on a Web server running on port 80 and set up the PGP 5.0 clients to access that port.

The remainder of this section describes how to set this up.

If you are setting up the PGP 5.0 clients to access the key server on port 80, an existing Web server may be able to also handle the HTTP gateway CGI scripts. The only requirements are that the machine must be running Solaris 2.5 or later or Windows NT and it must have TCP/IP access to the machine running the PGP Certificate Server.

To accomplish this, you must move two CGI scripts to a directory where your Web Server can run scripts, and an alias (or mapping) must be set up to point to the CGI scripts. The two scripts are listed below:

C:\Program Files\Network Associates\PGPcertd\web\cgi-bin\add.exe

C:\Program Files\Network Associates\PGPcertd\web\cgi-bin\lookup.exe

These CGI scripts must be accessible from a browser using the following URLs:

<http://<your.host.com>/pks/add>

<http://<your.host.com>/pks/lookup>

You may need to add an alias that maps `/pks/` to the directory where the CGI scripts reside. See your Web server's documentation for details on where to place CGI scripts and on how to create aliases.

The HTTP gateway requires a configuration file with the following name:

C:\<Windows directory>\PGPmit.cfg

The contents of the file should appear in the following format:

url ldap://<hostname>:<port number>

In the following example, the Server is running on the host `certserver.company.com` on port 389:

url ldap://certserver.company.com:389

If the port number is not included, it is assumed to be 389. You must also change the key server preference for the PGP 5.0 clients. In the PGP 5.0 client, the key server preferences must be changed to the hostname of the machine running the Web server, and to port 80.

If you want to use port 11371 and avoid reconfiguring PGP 5.0 clients, you may want to run a dedicated Web server process with a separate configuration file. Bind the server to port 11371 and follow the instructions above.

Configuring Internet Information Server 3 for use as an HTTP Gateway

IMPORTANT: This process requires the addition of site-global Default Document files. Understand the implications before you make the following changes to your installation.

1. Use Windows Explorer to open your installation directory and navigate to the `Web/CGI-BIN` directory. Create a new directory there named 'Add', and copy **Add.exe** into that new directory.
2. Using the Internet Service Manager, create a virtual directory `/pks` which points to `Web\CGI-BIN`. Disable all permissions to this directory.
3. Create a second virtual directory, `/pks/Add`, which points to `Web\CGI-BIN\Add`, to where you have previously copied **Add.exe**. Configure Read and Execute permissions for that directory.
4. Create a third virtual directory, `/pks/Lookup`, which points to `Web\CGI-BIN`. Configure Read and Execute permissions for that directory.

5. In the main server configuration, under the Default Document configuration, add **Lookup.exe** and **Add.exe** — in that order — comma-separated, to the list of default documents. For instance, if the Default Documents entry contains only *Default.htm* you'll update it to read:

Default.htm, Lookup.exe, Add.exe

IIS2 does not support the use of multiple Default Documents. Therefore, this configuration will not work under IIS2.

Administering the Server

When the Server is running, the Server responds to client requests and requires very little attention. The Administrative version of the PGP system software allows you to submit and retrieve certificates from the Server. In addition to adding and retrieving certificates, you can also disable or remove certificates from the Server.

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 TLS and LDAPS
- 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** configuration setting to control user access to the add, delete, and disable features.

When a key does not pass authentication, 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 re-submit them to the PGP Certificate Server. If the keys are invalid, you can delete them.

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 server that holds keys rejected by the key Server. To access the pending bucket, select the “Pending” box when performing a key search.

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

Importing and exporting keys

This section describes how to import and export keys.

Importing keys to the Certificate Server

When you set up your Server, 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 Certificate Server. Select **Start—>Programs—>PGP Certificate Server—>PGP Import**. Use the following import command to send all keys in a file to a Server:

```
pgpimport [-d] <file> ldap://<hostname>[:<port>]
```

Where *<file>* is the name of the file that you want to send to a Server, *<hostname>* is the name of the target Server, and *<port>* is the optional LDAP port number for the target Server. The LDAP port number is required only if the machine does not use the default LDAP port of 389. Use the *-d* option to delete the imported file after the import runs to completion.

Exporting keys from the Certificate Server

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

```
pgpexport [-v | -i | -l | -1] <directory> > <file>
```


The following paragraphs describe each of these switches:

Table 4-1. Switches for Export Command

Switch	Description
-v	Enables verbose mode. When verbose mode is enabled, the pgpexport command writes the following values to the standard error device, stderr: the key ID of each key you export, and a running total of the number of keys exported. By default, the verbose mode is turned off.
-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 Server database files are located. If you do not explicitly specify a directory, the current directory location is assumed.
<file>	By default, the exported output is sent to the standard output device. To save the keys to a file, you must redirect the output to a named file.
-l	pgpexport checks if the key database is already in use. If so, it aborts. To override this and have pgpexport run even if the database is in use, use the -l option.
-1	Indicates that the exported keyblock should be PGP Certificate Server version 1.x compatible. New features are removed from the keys before exporting the keys.

Using the Server's console

After you install the software, make the required configuration changes, and verify the new configuration, use the Server's console to perform the following tasks:

- Start and stop the Server (**Control** panel)
- Create the database and check configuration (**Control** panel)
- Change the Server's control port, secure port, or configuration file (**Control** panel)
- Monitor the server's activities (**Monitor** and **Events** panels)
- Disable or enable the Server's NT service function (**Command** panel)
- Identify the server's command port, IP address or hostname, and the console's command timeout value (**Command** panel)

The Control panel

Use the controls on the **Control** panel to perform the following tasks:

- ❑ **NOTE:** Before you start the Server, you can change the Server's port, secure port, and configuration file. The first time you run the console, the console displays the defaults for these settings.

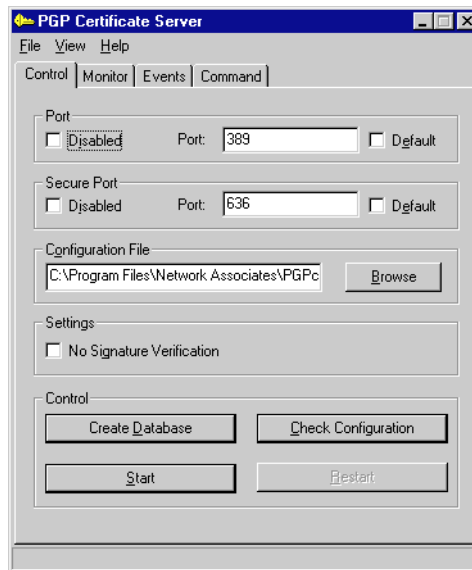


Figure 4-3. Server Console - Control Panel

- Identify the Server's port numbers (**Port**) - Use the default port or enter a different port address. When you check **Default**, the software uses the port number found in the configuration file; if the configuration file does not contain a port number, the software uses the default port 389 for the protocol in use. When you check **Disabled**, the port is disabled.
- Identify the Server's Secure port (**Secure Port**) - Use the default port or enter a different address. When you check **Default**, the software uses the port number found in the configuration file; if the configuration file does not contain a port number, the software uses the default port. When you check **Disabled**, the secure port is disabled.

- Identify the configuration file if it is different from the default. To change the configuration file for the NT service, use the **Browse** button (**Browse** button is active if the Server address is on the local machine), or enter the pathname in the **Configuration File** box. When the Server starts it saves the name of the new configuration file in the registry; the NT service uses the new registry setting when it automatically starts at boot time. Note that the configuration file specified here must reflect the path of the configuration file on the Server machine.

You can only modify the configuration file if NT Service **Disabled** is not checked.

- Create the Server's initial database or create a database in a new directory (Settings - **Create Database**) - Causes the Server to consult the configuration file's **Directory** setting to identify where to create new database files. The only time you must select this option is when you initially create your database or when you create a database in a new directory.
- Tell the Server not to perform signature verification (**No Signature Verification**) - Use this option when you are adding a large number of keys from one Server to another, and the signatures are known to be valid. This option speeds the process by eliminating signature checking (the Server assumes that all signatures are valid).
- Check the Server's configuration file for validity (**Check Configuration**) - Checks the validity of the values in the configuration file without starting the Server. If the program detects configuration problems, the console automatically advances to the **Events** panel and displays details about the problem. You may want to use this mode after you make configuration changes to make sure that all of the configuration values are valid. Note that any changes that you make to the configuration file do not take effect until the server is restarted.

You can also perform this task while the Server is running. However, changes to the configuration file do not take effect until the Server is restarted.

- Start the Server - When you click **Start**, the console opens a connection to the Server on the command port and sends a command to start the Server. If the Server is configured as an NT service, the console also sends commands to start the NT service. If the Server is already running when the console connects, the console displays the Server's current settings. The connected console displays three port numbers in its title bar - control port, secure port, and command port. If a port has been disabled, its port number is replaced in the title bar with the word "disabled."

- **Stop the Server** - The Console opens a connection to the Server on the command port and sends a command to stop the Server. If the Server is configured as an NT service, the console also sends commands to stop the service.
- **Restart the Server** - Restarts the Server using any new settings or configuration values.

If the Server is installed as an NT service, **Restart** stops the Server and starts it again. However, this action does not disconnect and then reconnect from the Server's command port, nor does it stop and start the service. Starting or restarting the Server causes the Server to reread its configuration file.

- **Server Help**

Note that when the Server is running, the settings on the **Control** and **Command** panels are grayed out (inactive).

The Command panel

Use this panel to define information about the Server running as an NT service or as a command line application.

Disabled (NT Service) - Check this box when the Server the console is to connect to is running as a command line application, or when you want to control only the Server functionality of an installed and running NT service, or if the Server is running on UNIX. When this box is checked, the console cannot start and stop the NT service.

Service Name - Displays the name of the NT service (that is, the name listed in the Services applet of the Windows Control Panel on the machine where the service is installed). Not user-configurable.

The following fields apply to both Server and NT service.

Name or IP Address - Identifies the IP address or name of the host where the NT service is installed or where the Server application is running. Use **127.0.0.1** or the string **localhost** for local host.

Command Port - Identifies the port number the server listens to for console connections and commands. The console will attempt to connect to this port.

Command Timeout - Directs the console to close the connection to the Server or NT service after a send or receive on the command port fails to complete after this length of time (in seconds). Also applies to NT service.

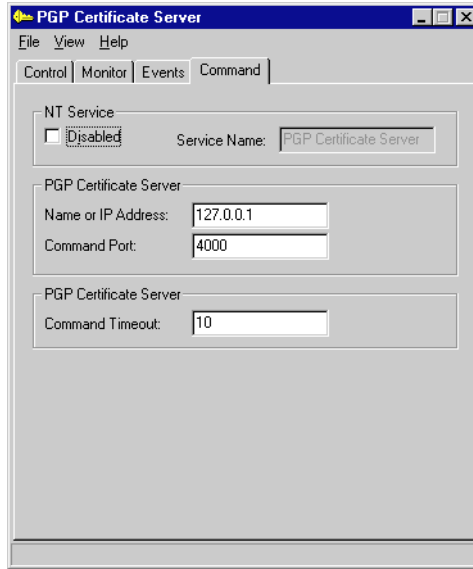


Figure 4-4. Server Console - Command Panel

The Monitor and Events panels

The console's Monitor and Events panel are described in [Chapter 7, "Monitoring and Logging."](#)

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

Certificate Servers are either Master Servers or Slave Servers.

- Master Servers perform all Server functions. A Certificate Server, replication log (a log where new and modified keys and destination Servers are recorded), and a Replication Engine reside on each master Server.
- Slave Servers perform all Server functions except adds, deletions, and disables. A Certificate Server resides on each slave Server.

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

To learn more about the replication process, see [“How the replication process works”](#) on page 85

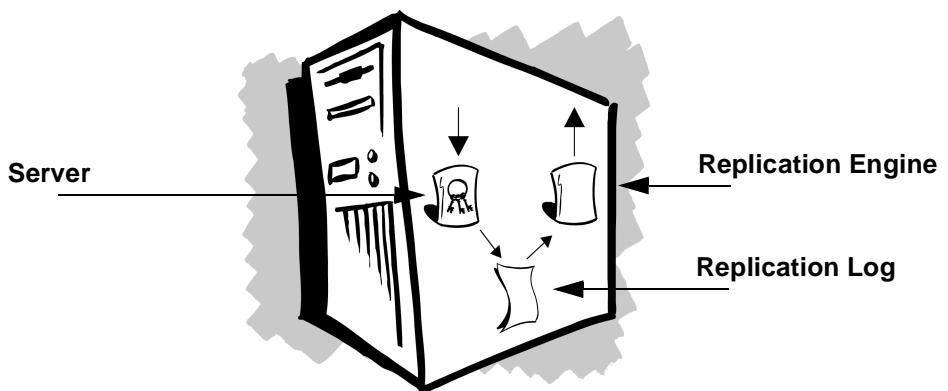


Figure 5-5. Certificate Server Components: Server, Replication Log, and Replication Engine

- ❑ **NOTE:** Requests for deletions and disables must be strongly authenticated using LDAPS or a Signed Request over LDAP. In either case, the Server 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 Servers must have the same `Allow keyid 0x???? delete` lines in their configuration files.
-

Server configurations

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

Server configuration models

The following Server configuration models represent only a few of the potential Server configurations. Each of the models identify the Server 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 Server 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 Server; if that Server goes down or has a problem, there is no backup. Tolerance is high, medium, or low.

- ❑ **NOTE:** When a Server 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 Server (Server A) with multiple slave or backup Servers (Server B, Server C, Server D, and so on). All Servers are available for user requests to update local keyrings, but only the master, Server A, can perform adds, deletions, and disables.

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

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.

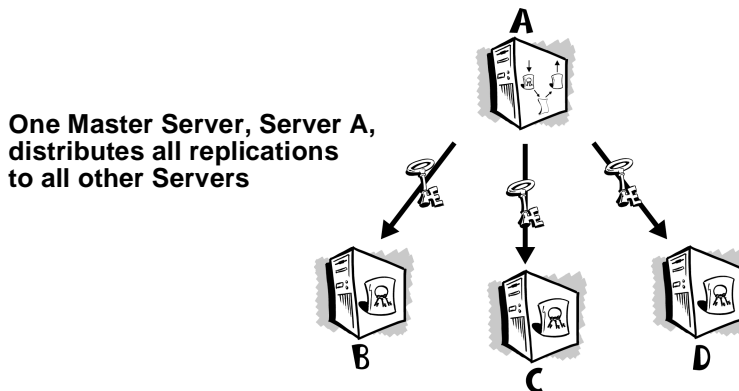


Figure 5-6. Master-Slave Model

Star model (medium efficiency, medium tolerance)

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

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

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

Let's look at an example. Server A is in New York, Server B is in Australia, Server C is in San Francisco, and Server 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 Server functions locally, but you must perform your adds, deletions, and disables remotely.

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

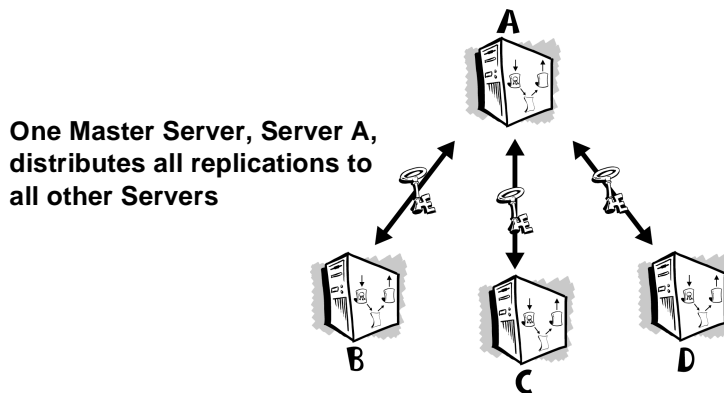


Figure 5-7. Star Model

Ring model (high efficiency, low tolerance)

This model is similar to the Star model. However, this model has a high efficiency rate. Users can perform adds, deletions, disables, and queries on all Servers.

The advantage to using this model rather than the Star model is that it limits the number of replications. If you do an add on Server B, Server B sends the add to Server C, Server C sends the add to Server D, Server D sends the add to Server A, and Server A sends the add to Server B. Server B recognizes that the key matches an existing key in its database, and does not replicate the add.

Each Server's configuration file has a replica line that identifies the Servers that receive replications from that Server. In this example, the replica line on each Server would contain the name of just one Server, the next Server in the ring.

Note that you can create hybrids of this model. For example, in addition to the existing replications, you could modify the configuration to include replications between Server A and Server C. You might also modify this configuration to include bidirectional replications between servers (for example, Server B replicates to Server C, and Server C replicates to Server B). This modification ensures that all servers are updated even if one link or server goes down.

Tolerance in this model is low because if any one of the Servers go down, replication cannot complete the loop to all Servers. Whereas in the Star model, if Server C goes down, Server A, Server B, and Server D continue to operate normally. Only Server C would be out of synchronization.

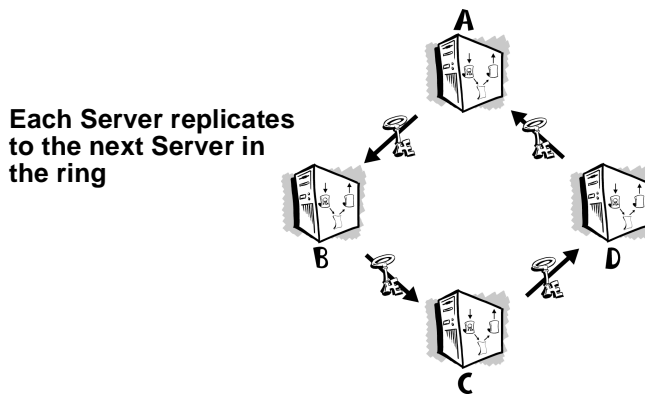


Figure 5-8. Ring Model

Fully connected model (low efficiency, high tolerance)

This model is similar to the Star and Ring models. You can perform all Server functions, including adds, deletions, and disables, on all of the Servers. The advantage in this model is that if any of the Servers go down, the other Servers continue to operate normally.

In the Star model, if System A goes down, System B, System C, and System D are on their own, and do not receive any replications during System A's down time. The Star model cannot give you 100% availability unless Server A is up all the time.

In the Fully Connected Model, if Server A goes down, Servers B, C, and D continue to work normally. The problem inherent in this model is that anything more than a small number of Servers becomes very inefficient. Each time you add a single key to a Server, you get multiple redundant replications (for example, if there are 4 servers, each add generates at least 12 replications). The more servers you add, the more replications you generate, and the less efficient this model becomes.

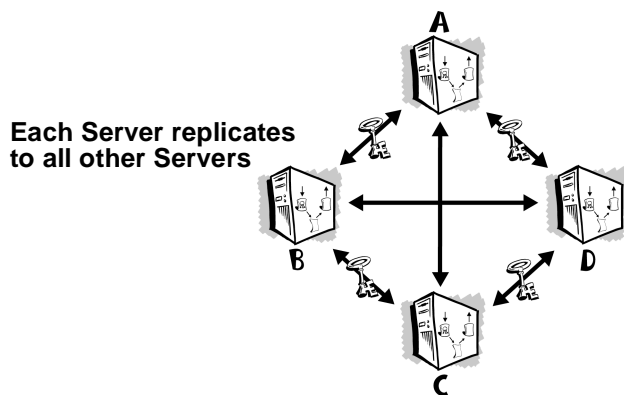


Figure 5-9. Fully Connected Model

Examples of different server configurations

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

Domestic company with one large location, four divisions - One Server is installed in each division and handles all Server requests that originate in that division. The Replication Engine on each Server replicates new or modified keys to the other three Servers. The four Servers are peers. This configuration is primarily used for load balancing.

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

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

How the replication process works

With the exception of Servers 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 Servers. Server A is the master server, and Server B and Server C are slave servers. The slave Servers, Servers B and C, can perform all Server functions except adds, deletions, and disables. The master Server, Server A, can perform all Server functions, including adds, deletions, and disables.

- When Server A receives a new or modified key, Server A checks the replica line in its configuration file to identify the Servers that it replicates or sends new or modified keys to. The names of Server B and Server C appear in Server A's configuration file.
- Server A places copies of the new or modified key in a queue (that is, its replication log or *repllog*) for Servers B and C.
- Server A's Replication Engine sees that there are keys in the queue, and looks for the machines that it must send the keys to. Server A finds the key designated for Server B, and sends it to Server B. Server A finds the key designated for Server C, and sends it to Server C. Since Servers B and C are slave Servers, they do not replicate the keys to other Servers.

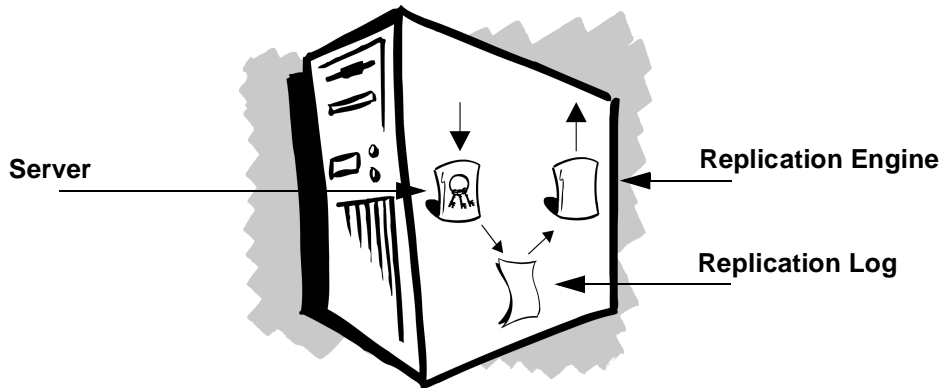


Figure 5-10. The Certificate Server's Components

When does replication occur?

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

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

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

How and where is the replication log (*replug*) maintained?

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

If all Servers are peers, each Server maintains a *replug* and database; the database is updated by the replication process.

Can two Servers replicate to each other?

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

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

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

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

If a Server is off-line, how and when is its database updated?

When a Server is off-line, the Replication Engine on other Servers continue to place new or modified keys for the target Server in the *relog*. When the Server is back on-line, the Replication Engines on the other Servers automatically update the Server's database.

If the master Server is off-line, databases are not updated until the master Server is back on-line.

Adding a new Server

To add a new Server to your configuration:

1. Install the Certificate Server software on the system that will run the new Server (see the *PGP Installation Guide* for details).
2. Edit the configuration files.
 - If the new Server is to replicate changes to one or more Server's, add the hostnames and optional port number of the Servers to the replica line of the new Server's configuration file.
 - If one or more Servers is to replicate changes to the new Server, add the new Server to the replica line of the configuration file for those Servers.
3. Export the existing database to the new Server.

☐ **NOTE:** Before you use the **pgpexport** command, stop Server A or configure it for Read Only mode.

- a. Login to the machine where the existing Server resides (Server A).
- b. Run **pgpexport**; supply the pathname of the directory that contains the current database (for example, C:\Program Files\Network Associates\PGPcertd\data). **pgpexport** creates an export file on Server A.
- c. Run **pgpimport** on Server A; specify ldap://<Server B>

For information on **pgpimport**, see [“Importing keys to the Certificate Server”](#) on page 72.

Learning about the Replication Engine

Engine and console compatibility

A engine can be connected to only one console at a time. The engine detects a console's version number when the engine and console connect over the command port. If the console version number is incompatible with the engine, the engine rejects the connection.

Console retrieves Monitor and Event data

The console retrieves Monitor data only if the console is connected to the engine and the engine is running. Event Log data is retrieved when the Events tab is selected while the console and engine are connected. The Events panel displays the most recent events.

Engine times out and closes connection

When the console and engine are connected over the command port, the engine closes the connection if the connection is inactive for a span of thirty minutes (that is, the console does not send a command to the engine during that time period). This feature gives other consoles a chance to connect to the engine.

Note that if **Auto Update** in the console's **Monitor** panel is set to request updates frequently and the Monitor panel is active, the engine does not time out and close the connection. In this situation, the console can monopolize the engine's command port, preventing another console from connecting with the engine.

Using the Replication Engine's console

Use the panels on the engine's console to perform the following tasks:

- Identify the engine's temporary file directory and the configuration and replication log files (**Control** panel).
- Start and stop the engine (**Control** panel).
- Monitor the engine's activity (**Monitor** panel).
- Review how well the engine is working (**Events** panel)
- Set NT Service and engine parameters (**Command** panel)

The engine's console runs on Windows NT. The engine's console and the engine must be compatible versions. If the console attempts to connect to an engine that is not compatible, the connection is rejected.

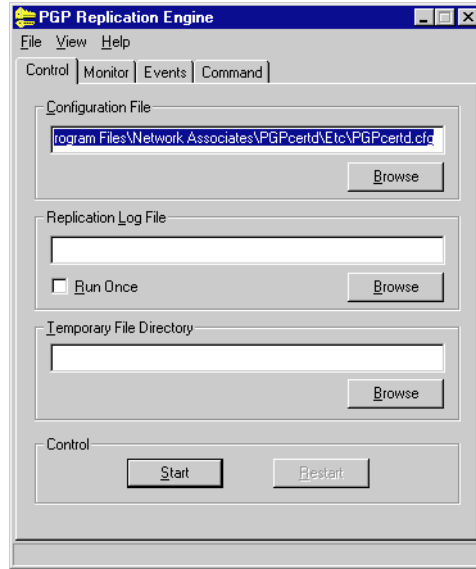


Figure 5-11. Engine Console - Control Panel

The Engine console's Control panel

Use the controls on the **Control** panel to perform the following tasks:

- Identify the configuration file you want the Replication Engine to use. By default, the Replication Engine uses the Certificate Server's configuration file, `pgpcertd.cfg`. This file contains all of the configuration settings that affect the Replication Engine. The configuration file is user-definable if you start the engine as an NT service from the console.
- Identify the replication log file you want the Replication Engine to use (contains a record of all changes to the database on the master Server). This option overrides a value in the configuration file. During normal operation, the Replication Engine continuously monitors the replication log file for more entries. When you select the "Run Once" option, the Replication Engine looks at the replication log file once only. The replication log file is user-definable.
- Identify the directory you want the Replication Engine to use as a temporary file directory. By default, the Replication Engine uses the values in the `TEMP` or `TMP` environmental variables to identify the temporary file directory. If these variables are not defined, the files are stored in the current directory. The temporary file directory is user-definable.

- **Start the Engine** - When you click **Start**, the console opens a connection to the engine on the command port and sends a command to start the engine. If the engine is configured as an NT service, the console also sends commands to start the NT service. If the engine is running when the console connects, the console displays the engine's current settings. The connected console also displays the command port number in its title bar. If a port has been disabled, its port number is replaced in the title bar with the word "disabled."
- **Stop the Engine** - The Console sends a command to stop the Server over the command port. If the Server is configured as an NT service, the console also sends commands to stop the service.
- **Restart the Engine** - If the Server is installed as an NT service, Restart stops the Server and starts it again. However, this action does not disconnect and then reconnect from the Server's command port, nor does it stop and start the service. Starting or restarting the Server causes the Server to reread its configuration file.

The Engine console's Command panel

Use this panel to define information about the engine running as an NT service or as a command line application.

Note that when you start the engine as an NT service, the values for **CommandPort** and **CommandTimeout** displayed on the engine console's **Command** panel are used as starting parameters for the NT service and override the NT service's registry settings and configuration file.

NT Service fields

Use these fields to define the engine's NT service.

Disabled - Check this box when the engine is running as a command line application, when you want to control only the engine functionality of an installed and running NT service, or if the engine is running on UNIX. When this box is checked, the console cannot start and stop the NT service.

If this box is checked, the console does not control the NT service. As a result, if you start the Server as an NT service and then check this box, you cannot stop the NT service by clicking **Stop**.

Service Name - Displays the name of the service (that is, the name listed in the Services applet of the Windows Control Panel on the machine where the service is installed). This is not user-configurable.



Figure 5-12. Engine Console - Command Panel

Name or IP Address and Command Port

Name or IP Address - Identifies the IP address or name of the host where the NT service is installed or where the engine application is running. Use **127.0.0.1** or the string **localhost** for local host.

Command Port - Identifies the port number the engine listens to for console connections and commands. If you start the engine as an NT service, this is the port number the engine uses, overriding the engine's settings in the registry and configuration file.

Command Timeout

Command Timeout - Directs the console to close the command port after a send or receive on the command port fails to complete after this length of time (in seconds). If you start the engine as an NT service, this timeout is also the timeout the engine uses, overriding its settings in the registry and configuration file.

The Engine console's Monitor and Events panel

See [Chapter 7, "Monitoring and Logging"](#) for details.

Setting Up Secure Mode

The Server includes a Secure Mode that you can use to perform deletions and other administrative tasks. When the Server is in Secure Mode, the Server 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.

The Server is shipped with public and private portions of a passphrase-less evaluation key for a default user ID. Use this key only for evaluation purposes. You must create your own key for the Server.

The following sections describe how to start the Server in Secure Mode using the Server evaluation key, and how to create your own key for the Server.

To start the Server in Secure Mode using the Server evaluation key:

1. Select the PGP Certificate Server from your system's Start menu (Start -> Programs -> PGP Certificate Server -> PGP Certificate Server Console). The Server displays its console and Control panel.
2. Deselect the **Disabled** check box in the **Secure Port** field on the Server's **Control** panel.
3. Edit your configuration file by changing Secure Mode to Required.
4. Press Start on the Server's **Control** panel. The Server displays a dialog box (**PGP Certificate Server: Enter passphrase for Secure Server Key**), with **Signing key** and **Passphrase of signing key** fields. The **Signing key** field displays the Server evaluation key (PGP Certificate Server Untrusted Evaluation Key), which does not require a passphrase.
5. Press **OK**. The titlebar for the Server's console displays the following text:

```
PGPcertd (running -389:636:4000)
```

389 is the regular Server port, and 636 is the secure Server port and 4000 is the command port. You are now in Secure Mode.

6. Stop the Server (click **Stop** on the console) and exit the Server's console (click the **X**, top right corner of the console).

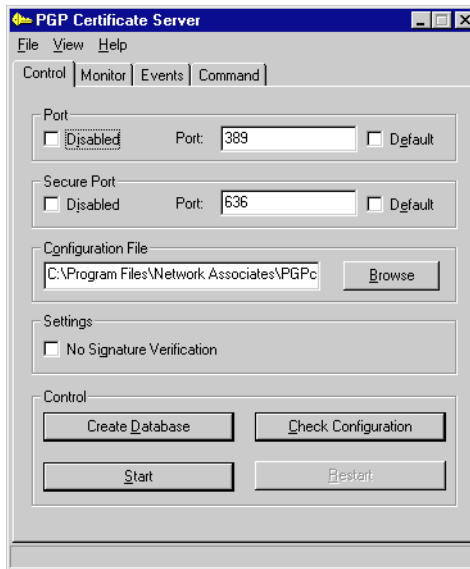


Figure 6-13. Server's Console - Control Panel

To create your own key, you must perform the following tasks:

- Display the Server's keyring in PGPkeys
- Create a new key for the Server
- Delete the evaluation key for the Server

NOTE: To create a new key, must have PGP version 6.5.1.

Display the Server's keyring in PGPkeys:

1. Select PGPkeys from your system's Start menu (Start -> Programs -> PGP -> PGPkeys).
2. Select **Preferences** from the PGPkeys **Edit** menu.
3. Select the **Files** tab. This panel displays your public and private keyring files.
 - If you are on a system dedicated to the PGP Certificate Server, go to [Step 4 on page 95](#) to set the Server's keyring as the default for PGPkeys.

- If you are on your own workstation, PGPkeys displays your personal keyring. In this case, follow the remaining steps in this section to display the Server's keyring in PGPkeys.

Record the values in these fields; you will need this information after you create the new Server keys. The normal path for your public and private keyring files, *pubring.pkr* and *secring.skr*, is as follows:

C:\Program Files\Network Associates\PGPnt\PGP Keyrings

4. To display the Server's keyring file in PGPkeys, do the following:
 - Click the **Browse** button in the **Public Key Ring File** field. PGPkeys displays the **Select Public Keyring File** file location box. Locate the Server's public keyring file, *PGPcertd-pubring.pkr*. The default directory for this file is as follows:

C:\Program Files\Network Associates\PGPcertd\etc

Click **OK**.
 - Click the **Browse** button in the **Private Key Ring File** field. PGPkeys displays the **Select Private Keyring File** file location box. Locate the Server's private keyring file, *PGPcertd-secring.skr*. The default directory for this file is as follows:

C:\Program Files\Network Associates\PGPcertd\etc

Click **OK**.
5. Click the **OK** button on the PGP Preferences screen. PGPkeys displays the Server's keyring. The only key on this keyring is the PGP Certificate Server Untrusted Evaluation Key. The following section tells you how to create a new key pair for the Server to replace the evaluation key.

Create a new key for the Server:

1. Select **New Key** from the PGPkeys Key menu. The Server displays the Key Generation wizard. Read the first screen and click **Next**.
Each of the following steps represents a new wizard screen.
2. Enter the full name and email address for the Server's new key pair. In the **Full name** field, enter *<company name> Certificate Server*, for example, Acme Certificate Server (the name can include spaces). In the **Email address** field, enter the email address for the Server administrator, for example, *admin@acme.com*. Click **Next**.

3. Select the type of key you want to generate. **Diffie-Hellman/DSS** is recommended and preselected. To generate a Diffie-Hellman/DSS key pair, click **Next**. To generate an RSA key pair, select **RSA** and click **Next**.
4. Select the size of the key pair you want to generate. **2048 bits** is recommended and preselected. To generate a 2048 bit key pair, click **Next**. To generate a different size key pair, select the size and click **Next**.
5. Select when you want the key pair to expire. **Key pair never expires** is preselected. It is recommended that you select **Key pair expires on** and select a date one or two years into the future. Click **Next**.
6. Enter a passphrase for the key pair's private key. The passphrase should be at least 8 characters long and should contain non-alphabetic characters. The Hide Typing box is checked. Enter a passphrase in the **Passphrase** field; enter the passphrase a second time in the **Confirmation** field. Click **Next**.

The wizard generates your new key pair. If you have selected a large key pair and you are on a slow machine, this process may take several minutes.

7. The wizard discusses sending your key to the Server. Leave the checkbox on this screen unchecked (**Send my key to the root server now**), and click **Next**.
8. Click **Finish**. The wizard adds the new key to the Server's keyring in PGPkeys. You have successfully created a new key for the Server.

Delete the evaluation key from the Server's keyring:

1. Delete the evaluation key (**PGP Certificate Server Untrusted Evaluation Key**) for the Server's keyring. To do so, select the key in the PGPkeys window, and select the **Delete** option from the PGPkeys Edit menu.

If you are on a system dedicated to PGP, you have completed the process.

If you are on your personal workstation, follow the steps below to display your own personal keyring in PGPkeys:

1. Select **Preferences** from the PGPkeys **Edit** menu.
2. Select the **Files** tab. This panel displays the Server's public and private keyring files.
3. To display your keyring file in PGPkeys, do the following:

- Click the **Browse** button in the **Public Key Ring File** field. PGPkeys displays the **Select Public Keyring File** file location box. Locate your public keyring file (you recorded this information earlier). The default directory for this file is as follows:

C:\Program Files\Network Associates\PGPnt\PGP Keyrings

Click **OK**.

- Click the **Browse** button in the **Private Key Ring File** field. PGPkeys displays the **Select Private Keyring File** file location box. Locate your private keyring file, *secring.skr* (you recorded this information earlier). The default directory for this file is as follows:

C:\Program Files\Network Associates\PGPnt\PGP Keyrings

Click **OK**.

4. Click the **OK** button on the PGP Preferences screen. PGPkeys displays your keyring.

Using Secure Mode

Use Secure Mode to perform administrative functions such as the deletion of keys. You can use Secure Mode for all interactions with the Server. For example, you can use Secure Mode to perform searches privately.

To see an example of Secure Mode, use the evaluation key shipped with the Server. For real Server installations, create a new key pair for the Server (see [page 93](#) for details). Thereafter, use the new key pair you create.

To use Secure Mode, follow the steps below:

1. Start the Server from the **Start** menu. The Server displays the **Control** panel.
2. Deselect the **Disabled** check box in the **Secure Port** field.
3. Click the Server's **Start** button. The Server displays the **PGP Enter Passphrase for Selected Key** window.
4. Enter the passphrase of the Server's signing key and click **OK**.

The Server is now in Secure Mode.

Secure Mode is required for remote console access.

This chapter describes how to monitor Server and Replication Engine activities and check the log files.

Monitoring Operations

While the Server or the Replication Engine is running, you can consult the **Monitor** and **Events** panels to find out how well the programs are performing. The following paragraphs describe these panels on the Server and the Replication Engine.

Monitoring the Server

The **Monitor** panel gives information about the keys submitted and retrieved from the Server (see [Figure 7-1 on page 101](#)). Use the **Auto Update** option on the **Monitor** panel to update this information automatically.

The **Events** panel gives useful information about the Server's performance ([Figure 7-2 on page 103](#)). This panel displays events that have occurred on the NT server since the Server was started.

On NT systems, events are retrieved from the NT Event Log. On UNIX, events are retrieved from a Server-generated event log file. The console also saves the last set of events it received from the remote Server (both NT and UNIX), in a local event log file. The contents of this file are displayed if the console is not connected to the Server.

For a more detailed look at the errors that occur, consult the NT Event Log. The NT Event Log lists Server errors as well as other errors that may have occurred on the machine.

For a more in-depth chronological view of all the requests the Server has processed, examine the Access Log File. The level of information is controlled by the **AccessLogDetails** setting in the configuration file.

Events Panel Not Displaying Information

If the **Events** panel is not displaying information, the NT event log may be full. To correct this problem, use the steps that follow.

To display information on the Events Panel:

1. Run the Event Viewer (Start -> Programs -> Administrative Tools -> Event Viewer).
2. Select **Log Settings...** from the Log menu.
3. In the **Change Settings for ... Log** field, select Application.
4. In the **Event Log Wrapping to...** field, select the Overwrite Events as Needed option.

Monitoring the Replication Engine

The **Monitor** panel gives statistics that tell you how the replication is progressing on each of the slave machines where you are replicating information.

The **Events** panel gives you useful information about an NT or UNIX Replication Engine's activity.

- NT - For a more detailed look at the errors that occur on an NT engine, consult the NT Event Log. The NT Event Log lists engine errors as well as other errors that may have occurred on the machine.
- UNIX - On UNIX machines, the engine creates a UNIX event file. The UNIX event file is deleted and a new one created whenever the engine is restarted.

The following sections describe how to use the consoles and wizard to monitor Server and Replication Engine activity, and identify the underlying sources of information that the software uses to report this information.

Monitoring Certificate Server Activity

To monitor Server requests and activities, click the **Monitor** tab on the Server console (see [Figure 7-1](#)).

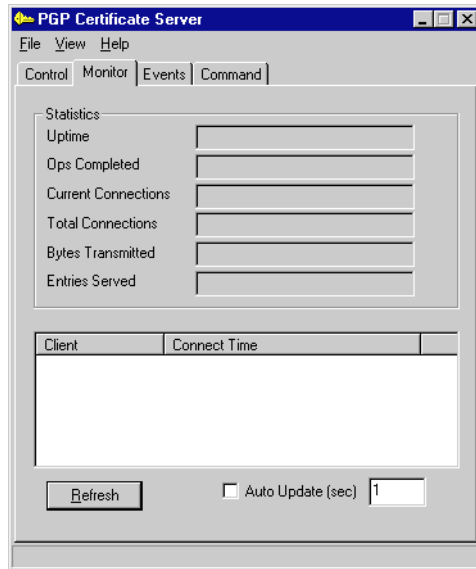


Figure 7-1. Server Console - Monitor Panel

Statistics

Uptime

Length of time the Server has been running.

Ops Completed

Number of client operations the Server has processed since the Server started the current run. This number is based on the following operations:

- Client connections opened
- Client connections closed
- Searches
- Adds

Current Connections

Number of active connections on the Server. Each connection consists of multiple operations. The number is typically low since most client connections are generally short-lived.

Total Connections

Total number of connections made to the Server since it was started. Under most circumstances, each connection is equivalent to one client session (for example, a client performing a search).

Bytes Transmitted

Number of bytes transmitted from the Server to the client since the Server was started.

Entries Served

Total number of certificates returned to clients through searches since the Server was started.

Client List

The lower section of the panel displays information on all clients currently connected to the Server. This information includes the hostname or IP address for each client, and the time and date the connection was established.

Monitoring Certificate Server Events and Errors

Use the **Events** panel on the console to see how the Server is performing (see [Figure 7-2](#)).

The **Events** panel displays all Server events and errors. Entries are ordered by their date and time. When the Events panel is selected, it displays all of the messages generated since the Server was last started. Click the Refresh button to update the listing.

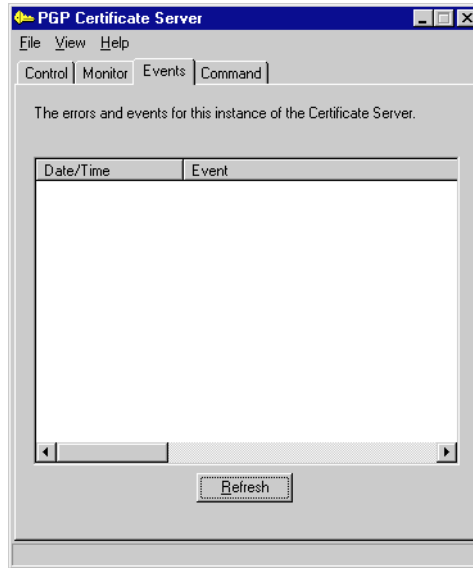


Figure 7-2. Server Console - Events Panel

Monitoring Replication Engine Activity

To view the statistics concerning the Replication Engine, click the **Monitor** tab on the Replication Engine's console (see [Figure 7-3](#)).

The **Monitor** panel displays the functions that the Replication Engine is processing. Information includes the address of the host machine and the port it is connected to. To update this information automatically, use the "Auto Update" option on the Monitor panel.

Status

Displays the current status for the Replication Engine: Up (Server available), Down (Server unavailable), or Untried (Server not tried).

Queue Size

Displays the number of replication operations that are currently lined up for processing.

Last Update Time

Displays the time that the last successful replication occurred for the selected host.

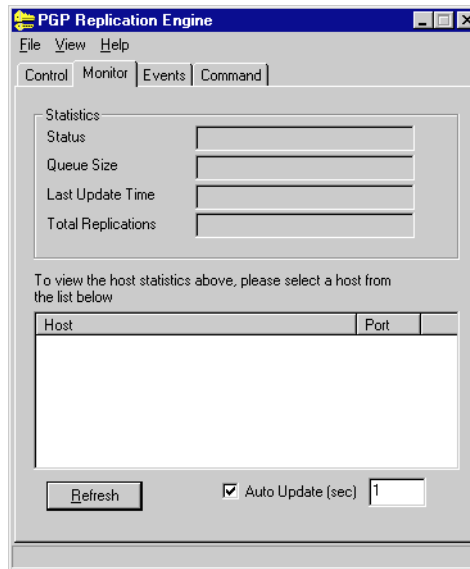


Figure 7-3. Replication Engine Console - Monitor Panel

Total Replications

Displays the total number of successful replications that have occurred since the Replication Engine was last started.

Monitoring the Replication Engine's Events and Errors

To see how well the Replication Engine is performing, click the **Events** tab from the console (see [Figure 7-4 on page 105](#)).

The **Events** panel displays all the Replication Engine's events and errors that have occurred since the Server was last started. They are ordered by date and time. Click the Refresh button to update the listing.

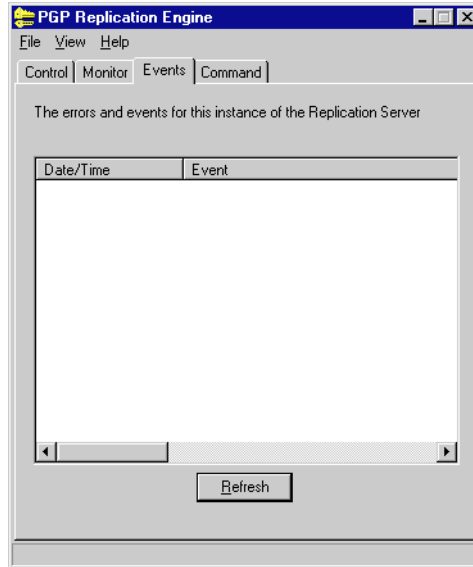


Figure 7-4. Replication Engine Console - Events Panel

Monitoring Remotely with the Wizard

Most event monitoring and analysis is performed from the console. To perform these functions remotely from another system, use the Web-based Configuration/Monitoring wizard.

To use the wizard:

1. Load the **Configuration/Monitoring wizard** with your preferred Web browser. The URL varies based on the location and port number of your Web Server, but should look like this:
`http://<hostname>[:<port>]/certserver/`
2. To check the status, click the **Status** tab or icon shown by the wizard. To monitor more than one Server, or to monitor a Server running on a different host, enter a new Server name and port number.
3. To see the information in the Access Log File, click the **Logs** tab.

The following sections describe the Access Log File and NT Event Log.

Examining the Access Log File

The Access Log File contains entries for each request that the Server has processed. The level of information in the log file is controlled by the “AccessLogDetails” configuration setting. Log entries appear in the following format:

operation session time result IP host type-specific

Table 7-1. Values in Log Entries

Value	Description
operation	A three letter code describing the type of request submitted to the Server. The following codes may appear in the Access Log File: BND = Bind operation UBD = Unbind operation ABN = Abandon SRC = Search operation ADD = Add certificate operation MOD = Modify entry operation DLT = Delete operation
session	A connection identifier that ties together multiple operations done over the same connection.
time	The time the request was generated. The time, given in Universal Time Coordinates (GMT), is expressed using the following format: YYYY-MM-DD HH:MM:SS
result	An LDAP result code in decimal format. For more information on the meaning of these codes, see the LDAP documentation.
IP	The IP address, in dotted decimal format, of the client making the request.
host	The host address of the client making the request. If the host address cannot be resolved, a dash appears in this field.

Table 7-1. Values in Log Entries

Value	Description
type-specific	<p>The type-specific information for the specified operation. The following are the returned values (values vary depending on the type of operation):</p> <p>BND The LDAP bind name enclosed in double quotes. A dash is used if the name cannot be resolved (normal case).</p> <p>UBD None</p> <p>ABN None</p> <p>ADD The ID of the certificate that was added to the Server. On signed requests, the ID of the certificate of the signer is also shown. A dash is used if there is no signer.</p> <p>MOD The ID of the modified certificate. On signed requests, the ID of the certificate of the signer is also shown. A dash is used if there is no signer.</p> <p>SRC The number, in decimal, for the matches or hits returned by a search operation, followed by a dash (-). Note that the trailing dash may not always be present.</p> <p>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</p>

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

Sample Access Log File Entries

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.nai.com 1 -  
SRC 0 1998-05-26 23:46:21 0 171.169.27.75 xyz.nai.com 3 -  
SRC 7 1998-05-26 01:16:03 0 171.169.27.75 xyz.nai.com 1 -  
ADD 7 1998-05-26 01:16:03 0 171.169.27.75 xyz.nai.com 0x27535B7C40C97D40 -  
SRC 0 1998-05-27 01:31:24 0 171.169.17.15 abc.nai.com 1 -  
SRC 0 1998-05-27 01:31:24 0 171.169.17.15 abc.nai.com 13 -
```

Access Log File Cycling

Periodically, the Server 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 Server operations.

The Access Log File settings in the configuration file control how frequently the Server cycles the entries in the Access Log File, and how many cycled files are retained on the Server. For more information, see [“General configuration settings” on page 35](#).

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, 1998, would be named cert.19980430.log. If the Access Log File name does not have an extension, the date becomes the extension. For example, cert.19980430.

Cycled files are kept in the same directory as the Access Log File. If the Server attempts to cycle data and a file with today’s date already exists, the Server 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 Server is controlled by the **CycleLogKeep** configuration setting. Each time the Server cycles, the Server 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 Server deletes enough old cycled files to satisfy the limit set by **CycleLogKeep**.

Note that the Server 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 “**CycleLogKeep <number> — Logs to Keep**” on page 36.

Examining the NT Event Log

In addition to the operational information provided in the Access Log File, other information is sent to the NT Event Log. The level of information that appears in the NT Event Log is controlled by the “**LogLevel**” configuration setting.

All entries in the NT Event Log that are associated with the Server have a source of “*pgpcertd*”. This distinguishes these messages from those generated by other processes.

LDAP Error Messages

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

Table 7-2. LDAP Error Messages in the Access Log File

LDAP Error Message	Description
0 (0x00) LDAP_SUCCESS	The request was successful.
1 (0x01) LDAP_OPERATIONS_ERROR	An unexpected Server error was encountered. See the Server Error Log for more information.
2 (0x02) LDAP_PROTOCOL_ERROR	The client accessing the Server is not following the proper protocol.

Table 7-2. LDAP Error Messages in the Access Log File

LDAP Error Message	Description
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 Server could not satisfy the entire request. You may need to contact a referral Server.
16 (0x10) LDAP_NO_SUCH_ATTRIBUTE	This requested attribute is not available.
17 (0x11) LDAP_UNDEFINED_TYPE	This error cannot be returned by the Server.
18 (0x12) LDAP_INAPPROPRIATE_MATCHING	This error cannot be returned by the Server.
19 (0x13) LDAP_CONSTRAINT_VIOLATION	Due to Server 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.
21 (0x15) LDAP_INVALID_SYNTAX	An invalid keyblock was received by the Server. See the Server's 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 Server.

Table 7-2. LDAP Error Messages in the Access Log File

LDAP Error Message	Description
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 Server.
36 (0x24) LDAP_ALIAS_DEREF_PROBLEM	This error cannot be returned by the Server.
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 Server.
51 (0x33) LDAP_BUSY	This error cannot be returned by the Server.
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 Server.
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 Server.
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 Server.

Table 7-2. LDAP Error Messages in the Access Log File

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 Server.
70 (0x46) LDAP_RESULTS_TOO_LARGE	This error cannot be returned by the Server.
80 (0x50) LDAP_OTHER	This error cannot be returned by the Server.
81 (0x51) LDAP_SERVER_DOWN	The client detected that the Server was not accessible. The host or port number may not be correct, the network may be having problems, or the Server 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 Server was incorrect or corrupted.
84 (0x54) LDAP_DECODING_ERROR	The data sent by the Server was incorrect or corrupted.
85 (0x55) LDAP_TIMEOUT	This error cannot be returned by the Server.
86 (0x56) LDAP_AUTH_UNKNOWN	This error cannot be returned by the Server.
87 (0x57) LDAP_FILTER_ERROR	This error cannot be returned by the Server.
88 (0x58) LDAP_USER_CANCELLED	The operation was aborted by the user.
89 (0x59) LDAP_PARAM_ERROR	The certificate received by the Server is invalid. The keyblock may be missing.
90 (0x5a) LDAP_NO_MEMORY	A memory allocation problem occurred.

Removing the Software

To remove the Server:

1. Use the Add/Remove Programs function from the Windows Control Panel.

-
- **NOTE:** The uninstall program does not delete all files from the Server directory (by default, the Server directory is C:\Program Files\Network Associates\PGPCertd). After you run uninstall, remove any unwanted files and directories.
-

Glossary

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	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.
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 key pairs 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.
key ID	A legible code that uniquely identifies a key pair. Two key pairs may have the same user ID, but they will have different Key IDs.
key pair	A public key and its complimentary private key. In public-key cryptosystems, like the PGP program, each user has at least one key pair.
keyring	A set of keys. Each user has two types of keyrings: a private keyring and a public keyring.
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.
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 key pair-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 key pair-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 key pair is used, and no security is needed in the channel itself.
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.
SLAPD	An LDAP implementation developed at the University of Michigan which defines the actual functions used to access information (certificates) from a centralized server.
SLURPD	An LDAP extension that allows the contents of a database to be replicated from master to slave servers.
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.
user ID	A text phrase that identifies a key pair. 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 key pair.
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

A

- a Server command line switch [58](#)
- accepting keys [19](#)
- access
 - controls, establishing [46](#)
 - Parameter (Allow Server configuration setting) [48](#)
- Access Log File [33](#)
 - cycling [108](#)
 - day archived (CycleLogDay) [33](#)
 - description [21](#)
 - description of entries [106](#)
 - examining [99](#)
 - LDAP error messages [109](#)
 - location of [33](#)
 - log entries [106](#)
 - naming conventions for cycled files [108](#)
 - number retained (CycleLogKeep) [33](#)
 - retention period for cycled files [109](#)
 - sample entries [108](#)
 - time of day archived (CycleLogTime) [33](#)
 - using the Configuration/Monitoring wizard to view [105](#)
- Access Permitted [48](#)
- AccessLogDetails [33, 47](#)
 - configuration setting [99, 106](#)
- AccessLogFile [33, 47](#)
- Action on Key Policy Failure [40](#)
- adding a Server [88](#)
- aliases [31](#)
 - Web documents [31](#)
- Allow [33, 47](#)
 - access by [47](#)
 - configuration setting [71](#)
- Allow KeyID
 - command [42, 47](#)
 - configuration setting [29, 44](#)

- Allow Server configuration setting [46](#)
- Allowed Certificate Signatures [39](#)
- AllowKeyid command [28](#)
- AllowSigID [33, 39](#)
 - certificate policy configuration setting [29, 40 to 41, 44](#)
- Apache Server 1.3 [30](#)
- ASCII-armored
 - key file, description [20, 29](#)
 - text, definition [115](#)
- authentication
 - definition [115](#)
 - engine [64](#)
- Auto Update [67](#)
 - Replication Engine Monitor panel [103](#)
 - Server Monitor panel [99](#)
- auto-start
 - configuring Server for [60](#)
 - invoking Server with [60](#)

B

- Bytes Transmitted
 - Server activity statistics [102](#)

C

- c Replication Engine command line switch [65](#)
- c Server command line switch [58](#)
- CacheEntries [33, 38](#)
- certificate
 - definition [18, 115](#)
- certificate policy
 - configuration matrix [41](#)
 - configuration settings [39](#)
- Certification Authority
 - definition [115](#)

- certify
 - definition [115](#)
- CGI
 - interface [18](#)
 - scripts [31](#), [68](#)
 - scripts, accessing [69](#)
- Check Configuration feature [30](#), [75](#)
- Client List
 - Server activity statistics [102](#)
- command line
 - starting Replication Engine [64](#)
- Command panel
 - engine [91](#)
 - Server console [76](#)
- Command Port
 - engine Command panel [92](#)
- Command Timeout [36](#)
 - engine console Command panel [92](#)
- CommandPort setting [28](#) to [29](#), [33](#), [35](#)
- CommandTimeout [28](#), [33](#), [36](#)
- configuration file
 - corrupted [32](#)
 - deleted [32](#)
 - editing [32](#)
 - for HTTP gateway, pgpmit.cfg [70](#)
 - pgpcertd.cfg [27](#), [32](#)
 - pgpcertd.conf [31](#)
 - Replication Engine [62](#) to [63](#), [90](#)
 - Server command line switch [58](#)

- configuration settings
 - AccessLogDetails [33, 47](#)
 - AccessLogFile [33, 47](#)
 - Allow [33, 47](#)
 - Allow KeyID command [42](#)
 - AllowSigID [33, 39](#)
 - brief descriptions [33](#)
 - CacheEntries [33, 38](#)
 - Certificate Policy [39](#)
 - changing [31](#)
 - CommandPort [33](#)
 - CommandTimeout [33](#)
 - CycleLogDay [33, 36](#)
 - CycleLogKeep [33, 36](#)
 - CycleLogTime [33, 36](#)
 - database [38](#)
 - DBCachSize [33, 38](#)
 - DefaultAccess [33, 50](#)
 - Directory [33, 38](#)
 - formatting rules [35](#)
 - general [35](#)
 - IdleSyncTimeout [33, 38](#)
 - LogLevel [33, 37](#)
 - Mode [33, 38](#)
 - MustSigID [34, 39](#)
 - PolicyFailures [34, 40](#)
 - Port [34, 37](#)
 - PrivateKeyRing [34, 44](#)
 - PublicKeyRing [34, 44](#)
 - RandSeedFile [34, 45](#)
 - ReadOnly [34, 38](#)
 - RepdCommandPort [34, 42](#)
 - RepdCommandTimeout [34, 42](#)
 - Replica [34](#)
 - Replication Engine [41](#)
 - ReplicationSecureKeyID [34, 43](#)
 - RepLogFile [34](#)
 - Secure Mode [34, 44 to 45](#)
 - SecurePort [34, 45](#)
 - ServerSecureKeyID [34, 45](#)
 - SizeLimit [35, 37](#)
 - TimeLimit [35, 37](#)
 - TrimPhotoIDs [35, 40](#)
 - TrimSigs [35, 40](#)
 - TrimUsers [35, 40](#)
 - verifying [30](#)
 - verifying validity [30](#)
 - Configuration/Monitoring wizard [21, 30](#)
 - description [19, 30](#)
 - URLs [31](#)
 - using [31](#)
 - configuring
 - Server console [28](#)
 - Server to auto-start [58, 60, 68](#)
 - the alias for Web documents [31](#)
 - the Replication Engine [27](#)
 - the Server [27](#)
 - console
 - and engine compatibility [88](#)
 - and Server compatibility [67](#)
 - Control panel [53, 74](#)
 - Check Configuration button [30](#)
 - engine [90](#)
 - Server console [74](#)
 - Start button [56](#)
 - conventional encryption
 - definition [115](#)
 - corrupted configuration file [32](#)
 - Create Database setting [75](#)
 - create new key for Server [95](#)
 - criteria used to accept or reject keys [19](#)
 - Current Connections
 - Server activity statistics [102](#)
 - CycleLogDay [33, 36](#)
 - CycleLogKeep [33, 36, 109](#)
 - CycleLogTime [33, 36](#)
- D**
- d
 - Replication Engine command line switch [65](#)
 - Server command line switch [59](#)

- database [34](#)
 - cache size [33](#)
 - file permissions [33](#)
 - location of files [33](#)
 - location on slave Servers [34](#)
- Database Access Mode [38](#)
- database configuration settings [38](#)
- Day to Cycle Log [36](#)
- DBCachSize [33, 38](#)
- debug
 - levels [59](#)
 - mode [59](#)
- decryption
 - definition [115](#)
- DefaultAccess [33, 50](#)
- delete authority [50](#)
- deleted configuration file [32](#)
- digital signature
 - definition [115](#)
- Directory [33, 38](#)
- directory export command line switch [73](#)
- disabling Secure Mode [58](#)

E

- efficiency [80](#)
- encryption
 - definition [115](#)
- engine
 - and console compatibility [88](#)
 - authentication [64](#)
 - closes connection [89](#)
 - restarting [91](#)
 - starting [91](#)
 - stopping [91](#)
 - times out [89](#)
- engine console
 - Command panel [91](#)
 - Control panel [90](#)
 - retrieving data [89](#)
 - using [89](#)

- establishing access controls [46](#)
- evaluation key
 - deleting [96](#)
 - Secure Mode [93](#)
- Event Viewer [100](#)
- events and errors
 - Server [102](#)
- Events panel [30, 60, 99, 102](#)
 - description [21](#)
 - not displaying information [100](#)
 - Replication Engine [61, 100, 104](#)
 - Replication Engine console [66](#)
- export command [72](#)
- export command line switches
 - directory [73](#)
 - file [73](#)
 - i [73](#)
 - l [73](#)
 - v [73](#)
 - 1 [73](#)
- exporting keys [20, 72](#)
- extracting Key IDs for configuration [29](#)

F

- f Replication Engine command line switch [65](#)
- f Server command line switch [58](#)
- file export command line switch [73](#)
- fingerprint
 - definition [115](#)
- Fully Connected Model [83](#)

G

- general configuration settings [35](#)

H

- Hosts to Replicate Database to [43](#)

- HTTP [43, 68](#)
 - gateway CGI scripts [69](#)
 - gateway configuration file, `pgpmit.cfg` [70](#)
 - gateway, setting up [69](#)
 - HTTP-to-LDAP gateway [68](#)
 - protocol [18](#)
 - support [68](#)
- I**
- i export command line switch [73](#)
- identifying directory and files for Replication Engine [62](#)
- IdleSyncTimeout [33, 38](#)
- import command [72](#)
- importing keys [20, 72](#)
 - PGP Import command [72](#)
- installation wizard
 - description [18](#)
- introducer
 - definition [115](#)
- invoking Server with auto-start option [60](#)
- Items to Log in Access Log [47](#)
- K**
- key escrow
 - definition [116](#)
- key fingerprint
 - definition [116](#)
- key IDs
 - definition [116](#)
 - extracting [29](#)
- key pair
 - definition [116](#)
- keyring [72](#)
 - definition [116](#)
 - description [20](#)
- keys
 - allowed when TrimSigs is on [33](#)
 - definition [116](#)
 - description [17](#)
- Keys Served
 - Server activity statistics [102](#)
- L**
- l export command line switch [73](#)
- Last Update Time
 - Replication Engine Monitor panel [103](#)
- LDAP [18, 20, 43, 68](#)
 - definition [116](#)
 - description [18](#)
 - error messages [109](#)
 - port [389](#) [72](#)
 - using search and retrieval functions [20](#)
- LDAP search attributes
 - creation and expiration dates [20](#)
 - email address [20](#)
 - key ID [20](#)
 - PGP key type, size, revocation status [20](#)
 - user name [20](#)
- LDAPS [43](#)
- level of access [33](#)
- Lightweight Directory Access Protocol
 - description [18](#)
- localhost [28](#)
- log entries [106](#)
- Logging Level [37](#)
- LogLevel [33, 37](#)
 - Server configuration setting [109](#)
- Logs to Keep [36](#)
- M**
- making configuration changes [31](#)
- mappings [31](#)
- master Servers [41, 79](#)
- Master Slave Model [80](#)

- message digest
 - definition [116](#)
- meta-introducer
 - definition [116](#)
- Microsoft's Internet Information Server 2.0 [31](#)
(IIS) [30](#)
- MIT-style key servers [69](#)
- Mode [33](#), [38](#)
- Monitor panel [21](#), [60](#)
 - Auto Update option [99](#)
 - Replication Engine [61](#), [100](#), [103](#)
 - Replication Engine console [66](#)
 - Server [99](#)
- monitoring
 - remotely [105](#)
 - Replication Engine activity [21](#), [100](#), [103](#)
 - Replication Engine events and errors [104](#)
 - Server activity [21](#), [99](#), [101](#)
 - Server events and errors [102](#)
- MustSigID [34](#), [39](#)
 - certificate policy configuration setting
[29](#), [39](#) to [41](#), [44](#), [46](#)
- N**
- n Server command line switch [58](#)
- Name or IP Address field, engine Command panel [92](#)
- Netscape's FastTrack Server 3.01 [30](#)
- Network Associates
 - contacting
Customer Care [xiii](#)
- No Signature Verification [75](#)
 - setting [55](#)
- NT Administrator access privileges [53](#), [61](#)
- NT Event Log [99](#), [105](#), [109](#)
 - description [21](#)
 - Replication Engine [100](#)
- NT service
 - stopping [57](#)
- NT Service fields
 - engine console Command panel [91](#)
- number of database entries cached by Server [33](#)
- O**
- o Replication Engine command line switch [65](#)
- O'Reilly Software's WebSite Professional 2.0 [30](#)
- Ops Completed
 - Server activity statistics [101](#)
- overriding settings in configuration file [29](#)
- P**
- p Server command line switch [58](#)
- passphrase
 - definition [116](#)
- passphrase-less evaluation key
 - Secure Mode [93](#)
- password authentication [71](#)
- pending bucket [34](#), [71](#) to [72](#)
- performance of Server and Replication Engine [21](#)
- performing administrative functions [97](#)
- Perl
 - script [30](#)
- persistent pseudo random seed [34](#)
- PGP Certificate Server
 - features of [17](#)
 - Untrusted Evaluation Key [96](#)
- PGP Import command [72](#)
- PGP keyring [29](#)
- PGP 5.0 [68](#)
- pgpcertd [109](#)
 - pgpcertd.cfg configuration file [27](#), [32](#), [65](#)
 - pgpcertd.conf configuration file [31](#)
 - PGPcertd-pubring.pkr [27](#)
 - PGPcertd-secring.skr [27](#)
 - pgpexport command [72](#), [88](#)

- pgpimport command [72, 88](#)
- pgpkeyid utility [29](#)
- PGPkeys [67](#)
 - displaying Server's keyring [94](#)
- PGPrepd, PGP Replication Engine [41, 64](#)
- plaintext
 - definition [116](#)
- PolicyFailures [34, 40](#)
- Port [34, 37](#)
- port
 - setting [74](#)
 - 11371 [69](#)
 - 389 [58, 70, 74](#)
 - 80 [69](#)
 - 8080 [31](#)
- port number for client requests and certificate submittals [58](#)
- preparing to start Replication Engine from console [62](#)
- private key
 - definition [116](#)
- private keyring
 - definition [116](#)
- PrivateKeyRing [27, 29, 34, 44 to 45](#)
- public key
 - definition [117](#)
- public keyring
 - definition [117](#)
- public-key cryptography
 - definition [117](#)
- PublicKeyRing [27, 29, 34, 44 to 45](#)
 - configuration setting [44](#)
- RandSeedFile [34, 45](#)
- read/write access [34](#)
- ReadOnly [34, 38](#)
- redundant replications [80](#)
- Refresh button
 - Events panel [102](#)
 - Replication Engine Events panel [104](#)
- rejecting keys [19](#)
- related reading [xiv](#)
- Remove PhotoIDs [40](#)
- Remove Unallowed Signatures [40](#)
- Remove Unallowed User IDs [40](#)
- removing
 - Server software [27](#)
 - the software [113](#)
- RepdCommandPort [34, 42](#)
- RepdCommandTimeout [34, 42](#)
- Replica [34, 43, 85](#)
- replication
 - how it works [85](#)
 - of the database to other Servers [21](#)
 - Server offline [87](#)
 - Servers replicating to each other [87](#)
 - when it occurs [86](#)

Q

- Queue Size
 - Replication Engine Monitor panel [103](#)

R

- r Replication Engine command line switches [65](#)

- Replication Engine 18
 - configuration file 63, 90
 - configuration settings 41
 - configuring 27
 - description 21
 - errors 104
 - events 104
 - Events panel 100, 104
 - Monitor panel 100, 103
 - NT Event Log 100
 - performance 21
 - preparing to start from console 62
 - replication log file 63, 90
 - running the console 61
 - starting from command line 64
 - starting from console 64
 - stopping from console 66
 - Temporary File Directory 63, 90
 - using 79
 - Replication Engine command line switches
 - c 65
 - d 65
 - f 65
 - o 65
 - r 65
 - s 65
 - t 65
 - Replication Engine configuration settings
 - Replica 43
 - ReplicationSecureKeyID 43
 - RepLogFile 43
 - Replication Engine console
 - Events panel 66
 - Monitor panel 66
 - Start button 66
 - starting 61
 - Stop button 66
 - replication log
 - maintenance 86
 - Replication Log File 42 to 43, 63, 90
 - ReplicationSecureKeyID 34, 43
 - replug
 - maintenance 86
 - RepLogFile 34, 43
 - required
 - parameters for Server 27
 - Required Certificate Signatures 39
 - required Server parameters for console connection 28
 - restarting the engine 91
 - restricting access to the Server 71
 - retention period for cycled Access Log Files 109
 - retrieving
 - data
 - engine console 89
 - keys 20
 - revoked keys 41
 - Ring Model 82
 - Run Once 90
 - running
 - multiple Servers on same machine 60
 - Replication Engine console 61
 - the PGP Certificate Server console 53
- S**
- s
 - Replication Engine command line switch 65
 - Server command line switch 58
 - search attributes 20
 - secure access via TLS and LDAPS 71
 - Secure Mode 58
 - configuration settings 44
 - setting up 93
 - using 97
 - using Server evaluation key 93
 - Secure port setting 74
 - SecureMode 29, 34, 45
 - SecurePort 28, 34, 45

- Server
 - adding [88](#)
 - and console compatibility [67](#)
 - auto-start feature [68](#)
 - closes connection [67](#)
 - create new key [95](#)
 - delete evaluation key [96](#)
 - directory [113](#)
 - events and errors [102](#)
 - Events panel [102](#)
 - performance [21](#)
 - required parameters [27](#)
 - times out [67](#)
- Server activity statistics
 - Bytes Transmitted [102](#)
 - Client List [102](#)
 - Current Connections [102](#)
 - Keys Served [102](#)
 - Ops Completed [101](#)
 - Total Connections [102](#)
 - Uptime [101](#)
- Server command line switches [58](#)
 - a [58](#)
 - c [58](#)
 - d [59](#)
 - f [58](#)
 - n [58](#)
 - p [58](#)
 - s [58](#)
 - t [58](#)
- server configuration models [80](#)
 - Fully Connected Model [83](#)
 - Master Slave Model [80](#)
 - Ring Model [82](#)
 - Star Model [81](#)
- server configurations [80](#)
- Server console [73](#)
 - Command panel [76](#)
 - configuring [28](#)
 - Control panel [74](#)
 - starting [53](#)
- Server's
 - keyring, displaying in PGPkeys [94](#)
 - TLS key [68](#)
- Servers
 - running multiple on same machine [60](#)
- ServerSecureKeyID [28](#) to [29](#), [34](#), [45](#)
- setting up
 - Secure Mode [93](#)
 - the Configuration/Monitoring wizard [30](#)
- settings
 - Check Configuration [75](#)
 - Create Database [75](#)
 - No Signature Verification [75](#)
 - Port [74](#)
 - Secure port [74](#)
- sign
 - definition [117](#)
- signature
 - authentication [71](#)
 - definition [117](#)
- SizeLimit [35](#), [37](#)
- SLAPD
 - definition [117](#)
- slave Servers [21](#), [41](#), [79](#)
- SLURPD
 - definition [117](#)
- SSL [44](#), [93](#)
- Star Model [81](#)
- Start button
 - Replication Engine console [66](#)
 - Replication Engine Control panel [64](#)
 - Server Control panel [56](#)
- starting
 - Replication Engine console [61](#)
 - the engine [91](#)
 - the Replication Engine from command line [64](#)
 - the Replication Engine from the console [64](#)
 - the Server's console [53](#)

- starting Server 55
 - as NT service from console 56
 - from command line 57
 - starting Server console
 - from command line 54
 - from Start menu 53
 - Statistics
 - Server activity 101
 - Status
 - Replication Engine Monitor panel 103
 - Status tab
 - Configuration/Monitoring wizard 105
 - Stop button
 - Replication Engine 66
 - Replication Engine console 66
 - Replication Engine Control panel 64
 - Server Control panel 56
 - stopping
 - NT service 57
 - the engine 91
 - the Replication Engine from console 66
 - submitting keys 19
 - System Administrator authority level 20
 - System Log File 33
- T**
- t Replication Engine command line switch 65
 - t Server command line switch 58
 - t -1 Server command line switch 58
 - TCP/IP access 69
 - technical support
 - email address [xiii](#)
 - information needed from user [xiii](#)
 - online [xiii](#)
 - TEMP environment variable 63, 90
 - temporary files directory
 - Replication Engine 62 to 63, 90
 - text
 - definition 117
 - Time to Cycle Log 36
 - TimeLimit 35, 37
 - TLS 44, 93
 - key 34, 68
 - TMP environment variable 63, 90
 - tolerance 80
 - Total Connections
 - Server activity statistics 102
 - Total Replications
 - Replication Engine Monitor panel 104
 - Transport Layer Security (TLS) 44, 93
 - TrimPhotoIDs 35, 40
 - TrimSigs 35, 40
 - certificate policy configuration setting 41
 - TrimUserID 41
 - TrimUsers 35, 40
 - trusted
 - definition 117
 - trusted introducer
 - definition 117
- U**
- uninstalling the software 113
 - Uptime
 - Server activity statistics 101
 - URLs
 - Configuration/Monitoring wizard 31
 - user ID
 - definition 117
 - using
 - engine console 89
 - LDAP search function 20
 - Secure Mode 97
 - the Configuration/Monitoring wizard 31, 105
- V**
- v export command line switch 73
 - v export command switch 73

- verification
 - definition [117](#)
- verify configuration settings [30](#)
- verifying
 - configuration settings [30](#)
 - that the Replication Engine is running [66](#)
 - that the Server is running [60](#)

W

- Web documents [31](#)
- web of trust
 - definition [118](#)
- Web server [31](#)
 - using servers other than Microsoft IIS [31](#)
- who Parameter (Allow Server configuration setting) [48](#)
- Windows executable file [30](#)
- wizard
 - setting up [30](#)

Numerics

- 1 export command line switch [73](#)
- 127.0.0.1 [28](#)
- 32 or 64-bit key IDs [39](#)
- 64-bit key IDs [29](#)

