MICRO FOCUS

User Guide

Reflection PKI Services Manager

version 1.3.2

Legal Notice

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, see https://www.microfocus.com/about/legal/.

Third-Party Notices. Third-party copyrights and notices, including license texts and other materials passed through in compliance with third-party license terms, can be found in the `thirdpartynotices.txt` file in the program installation folder.

Contents

Reflection PKI Services Manager Features 5

1 Installing PKI Services Manager 7

System Requirements . . . 7
Windows Install and Uninstall . . . 7
Advanced Tab . . . 8
UNIX Install and Uninstall . . . 9
Upgrading From Earlier Versions . . . 10
PKI Services Manager Initialization . . . 11

2 Getting Started 13

PKI Services Manager Overview . . . 13
Configuration on Windows Systems . . . 14
- Start and Stop the PKI Services Manager Service on Windows . . . 14
- Configure PKI Services Manager on Windows . . . 15
- Save, Reload, and Restart on Windows . . . 16
- Check Validity and Mapping on Windows . . . 17

Configuration on UNIX Systems . . . 18
- Start and Stop the Service on UNIX . . . 18
- Configure PKI Services Manager on UNIX . . . 19
- Save, Reload, and Restart on UNIX . . . 20
- Check Validity and Mapping on UNIX . . . 21

3 Files and Application Data 23

Certificate Storage . . . 23
PKI Services Manager Public and Private Key . . . 24
PKI Services Manager Data Directories . . . 24
Change the Data Folder . . . 25
Windows Files . . . 26
UNIX Files . . . 27

4 PKI Services Manager Administration 29

Ensuring PKI Services Manager Availability . . . 29
Using a Server Cluster . . . 30
Configure a PKI Services Manager Cluster . . . 30
Configure Connections via a SOCKS Proxy . . . 32
Changing the JRE . . . 33

5 PKI Services Manager Console 37

Console Menu Commands . . . 37
- Set Data Folder Dialog Box . . . 38
- Test Certificate Dialog Box . . . 38
- Public Key Details Dialog Box . . . 39

General Pane . . . 39

Local Store Pane 41
Trusted Chain Pane 41
Add Trust Anchor 42
Local Store Browser 43
Windows Certificate Browser 43
Edit Trust Anchor 44
Clone Trust Anchor 44
Specify URI for Intermediate Certificate 44
Revocation Pane 45
Specify URI for CRL Server 45
Specify URI for OCSP Responder 46
Add OCSP Certificate 46
Revocation Settings 46
Identity Mapper Pane 47
Add Mapper Rule 48
Fetch Certificate 50

6 Troubleshooting 51

Troubleshooting PKI Services Manager Configuration 51
Troubleshooting Identity Mapping 51
Logging 52

7 Appendix 55

winpki and pkid Command Reference 55
pkid_config Configuration File Reference 58
pki_mapfile Map File Reference 62
Sample Mapping Rules 67
Sample Map File with RuleType Stanzas 68
pki-client Command Line Utility 68
PKI Services Manager Return Codes 71
DOD PKI Information 72
Certificate Attribute Requirements Enforced by PKI Services Manager 78

Glossary of Terms 81

Reflection PKI Services Manager Features

Reflection PKI Services Manager provides a service for validating X.509 certificates. You can configure supported Micro Focus products to use PKI Services Manager to validate certificates presented for authentication. PKI Services Manager can be installed on Windows or UNIX systems, and a single installation can support validation queries from multiple supported product installations. This user guide provides detailed information about PKI Services Manager. For additional information about configuring supported products to communicate with PKI Services Manager, refer to the product documentation.

Using Reflection PKI Services Manager you can:

- Centralize configuration and management of PKI services.
- Specify which certificates should be designated as the trust anchor when validating certificates presented by authenticating parties. On Windows systems, these can be certificates in the Windows system store.
- Configure access to intermediate certificates stored locally or on an LDAP or HTTP server.
- Configure revocation checking using CRLs stored locally or on an LDAP or HTTP server.
- Configure revocation checking using OCSP.
- Use flexible mapping criteria to determine which users or computers are allowed to authenticate with which certificates.
- Configure custom trust chain, revocation, and mapping settings for individual trust anchors.
- Maintain audit logs.
- Troubleshoot using debug logs.
- Enforce Federal Information Processing Standard (FIPS) 140-2 security requirements.
- Enforce United States Department of Defense PKI requirements.

1 Installing PKI Services Manager

In this Chapter

- “System Requirements” on page 7
- “Windows Install and Uninstall” on page 7
- “Advanced Tab” on page 8
- “UNIX Install and Uninstall” on page 9
- “Upgrading From Earlier Versions” on page 10
- “PKI Services Manager Initialization” on page 11

System Requirements

Reflection PKI Services Manager provides X.509 certificate validation services for supported Micro Focus products. For a list of products that Reflection PKI Services Manager, see Technical Note 2716 (http://support.attachmate.com/techdocs/2716.html).

After installing and configuring PKI Services Manager, you should configure your installed Micro Focus product to use certificates for authentication and to connect to PKI Services Manager. For details, search on "PKI Services Manager" in the product documentation.

Supported platforms

For information about supported platforms, see Technical Note 2427 (http://support.attachmate.com/techdocs/2427.html).

Console requirements

NOTE: The console provides a user interface for PKI Services Manager on Windows systems. The console is not required for configuring or running PKI Services Manager. You can use the commands and configuration files described in the Reference section (page 55) of this guide on all supported systems.

Requirements for running the console are:

- PKI Services Manager must be installed on a Windows system. The console is not supported on UNIX systems.
- The console requires a minimum display size of 800x600.
- To view the product Help, a supported Internet browser (such as, Internet Explorer, Firefox, or Google Chrome) is required. In addition, JavaScript must be enabled in the browser settings to navigate and search Help.

Windows Install and Uninstall

Reflection PKI Services Manager is available as a separate download at no additional charge when you purchase supporting products.

To install Reflection PKI Services Manager

1. Log in as an administrator.
2. Start the Setup Program (`Setup.exe`). If you are installing from the download site, the following steps start this program:
   - **2a** From the download site, click the Windows download link and run the download program.
   - **2b** Select a location for the installer files, and then click **Next**. This extracts the files to the specified location and starts the Setup Program.
3. Accept the default settings on the **Advanced** tab. (Creating an administrative installation image does not actually install the product — instead, it places the install files on a network location for later installation to multiple workstations.)
4. Start the service (page 14).

NOTE

- On Windows, starting the console or the service for the first time initializes (page 11) PKI Services Manager. This creates the required data folders and default settings files. If these folders already exist, they are not changed; PKI Services Manager uses your existing data files and folders. (On UNIX the install script automatically initializes PKI Services Manager if required, and starts the service.)
- Before Reflection PKI Services Manager can validate certificates you need to edit the default configuration and map files.

To uninstall Reflection PKI Services Manager

1. Log in as an administrator.
2. From the Windows **Programs and Features** (or the Add or Remove Programs) control panel, select Reflection PKI Services Manager.
3. Click **Uninstall** (or **Remove**).

Advanced Tab

Use the **Advanced** tab of the installer only if you want to modify the installer log settings or you are an administrator configuring a deployment.

Install to this PC	Installs PKI Services Manager to your computer.
Create an Administrative install image on a server	**NOTE:** An administrative install image does not actually install the product — instead, it creates an installation image that administrators can use to deploy PKI Services Manager to end users. When you create an administrative install image, an image of PKI Services Manager is copied to a network location for later installation to multiple workstations. This network location can be used by deployment tools to access and create packages that are deployed to workstations. Also, end users can perform installations by running `setup.exe` from this location.

Log file settings

By default, an installation log file is created and then deleted after installation successfully completes. (This configuration avoids accumulation of large log files after successful installations.) To save a log file for all installations, including successful ones, select **Create a log file for this installation**, and clear **Delete log file if install succeeds**.

The installation log file, which provides details about the installation, is saved in the user's Windows temporary folder (`%tmp%`) with a generated name that begins with `atm`. To open this directory, launch the **Start** menu **Run** command and enter `%tmp%`.

UNIX Install and Uninstall

Reflection PKI Services Manager is available as a separate download at no additional charge when you purchase supporting products.

To install Reflection PKI Services Manager

1. Log in as root.
2. Copy the installation package file to your computer and navigate to the directory that contains this file.
3. Use gzip to unzip the package:

   `gzip -d` *`package_name.`*`tar.gz`

   For example:

   ```
   gzip -d pkid_1.3.0.999-i386-solaris.gz
   ```
4. Use tar to expand the file:

   `tar -xf` *`package_name`*`.tar`

   This creates a directory based on the package name. For example:

   ```
   pkid_1.3.0.999-i386-solaris/
   ```
5. Change to this directory. For example:

   ```
   cd pkid_1.3.0.999-i386-solaris
   ```
6. Run the install script:

   ```
   ./install.sh
   ```
7. You are prompted to specify installation locations. To accept the default locations (recommended), press Enter in response to these prompts.

NOTE

- On UNIX the install script automatically starts the service.
- Before Reflection PKI Services Manager can validate certificates you need to edit the default configuration and map files.

To uninstall

1. Log in as `root`.
2. Run the uninstall script. This script is installed to the bin directory in the PKI Services Manager data folder. The default path is:

   ```
   /opt/attachmate/pkid/bin/uninstall.sh
   ```

NOTE: The uninstall script renames your existing configuration directory (`/opt/attachmate/pkid/config/` by default) using a name based on the current date, and time. For example, `config.20140101143755`. Your `local-store` directory and any certificates you have added to this directory remain unchanged.

Upgrading From Earlier Versions

Before upgrading a running copy of PKI Services Manager, review the upgrade procedure for your operating system.

To upgrade on Windows

1. You can install over your existing copy of PKI Services Manager.

   NOTE: If the PKI Services Manager service is running when you start the installation, the installer stops the service. Certificate validation services are not available while the service is stopped.

2. Start the service (page 14) after the installation is complete.

After the upgrade, PKI Services Manager uses your previously existing configuration. Your certificate store, revocation settings, identity mappings, and all other settings continue to work as they did prior to the upgrade.

To upgrade on UNIX

1. Uninstall (page 9) your existing copy of PKI Services Manager.

   The uninstall script renames your existing configuration directory (`/opt/attachmate/pkid/config/` by default) using a name based on the current date, and time. For example, `config.20140101143755`. Your `local-store` directory and any certificates you have added to this directory remain unchanged.

2. Install (page 9) the upgrade.

   The installer automatically starts the service. At this point, the service is running with a default configuration and a newly installed key pair. The next steps describe how to restore your prior settings and key pair using the backup configuration directory.

3. To restore your prior identification key, configuration settings, and mappings, you should stop the service. You can then replace the new default `config` directory with the backup copy and restart the service. For example:

   ```
   /etc/init.d/pkid stop
   cd /opt/attachmate/pkid
   mv config config_default
   mv config.20110101143755 config
   /etc/init.d/pkid start
   ```

PKI Services Manager Initialization

PKI Services Manager initialization depends on your operating system:

- On Windows systems, initialization happens after installation when you do any of the following: start the console, start the service, restart Windows, or use the initialization option of the **winpki** command line utility.
- On UNIX systems, initialization happens automatically when you run the install script.

What happens during initialization?

- User data folders (`config`, `logs`, `cache`, `local-store`, `temp`) are created in the PKI Services Manager data folder (page 82).
- Default `pki_config` and `pki_map` files are created in the `config` folder.
- Private and public keys are created in the `config` folder. These keys are used to verify the identity of the server to applications using the PKI Services Manager services.
- Correct folder and file permissions are set on files and folders.
- (Windows only) If an `sshd2_config` file is present from a Reflection for Secure IT server (version 6.1 or older) or an F-Secure server, settings for handling certificate authentication are migrated to PKI Services Manager configuration and map files. (On UNIX systems, you can manually migrate settings using the **pkid -m** option.)

2 Getting Started

In this Chapter

- “PKI Services Manager Overview” on page 13
- “Configuration on Windows Systems” on page 14
- “Configuration on UNIX Systems” on page 18

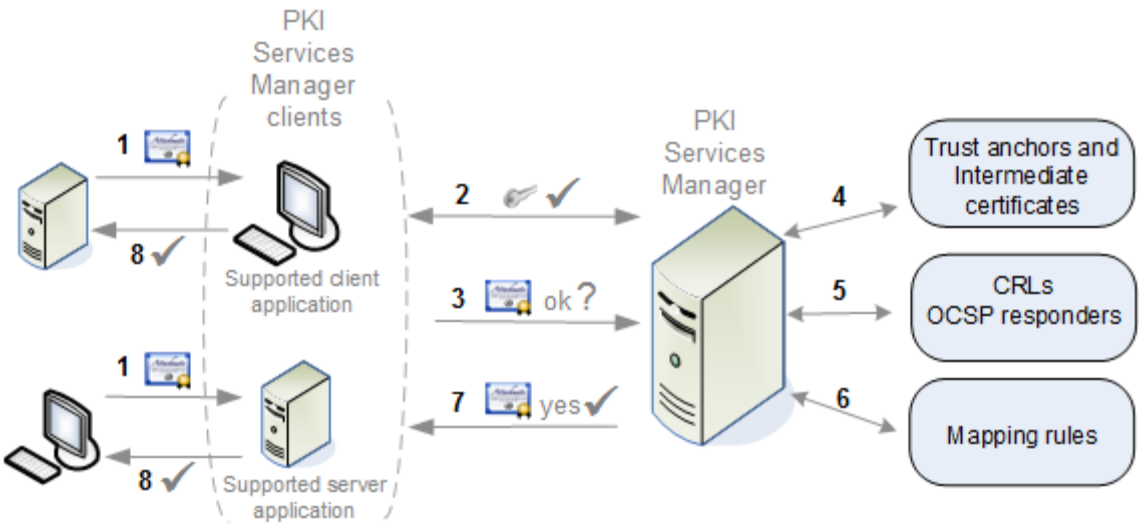
PKI Services Manager Overview

Reflection PKI Services Manager provides certificate validation services. One or more centrally managed installations of PKI Services Manager can provide certificate validation services for multiple Micro Focus applications.

Applications that use PKI Services Manager for certificate validation are referred to in this guide as PKI Services Manager clients. A PKI Services Manager client can be a Micro Focus client application authenticating a server host or a Micro Focus server application authenticating a client user. For example:

- A Reflection for Secure IT UNIX or Windows server verifying a certificate presented by an SSH client.
- A Reflection for Secure IT UNIX client verifying a certificate presented by an SSH server.
- A Reflection X Advantage session verifying a certificate presented for authentication by an X application host.
- A web-based Reflection Security Gateway or Reflection for the Web session that is configured to support TLS 1.2 certificate validation.
- The pki-client command line utility, which is provided with PKI Services Manager for testing certificate validation.

How it Works



1. During the authentication portion of the connection process, a server host or client user sends a certificate to a Micro Focus application (the PKI Services Manager client application). Before authentication can continue, the Micro Focus application needs to know that the certificate is valid and can be used for authentication by this host or client.
2. The client application connects to PKI Services Manager and uses an installed public key to authenticate the PKI Services Manager server.
3. The client application sends the certificate to PKI Services Manager.
4. PKI Services Manager checks that the certificate has not expired, is valid for the current use, and meets all attribute requirements (page 78). If these conditions are met, it verifies the chain of trust using your configured trust anchors and available intermediate certificates.
5. If required by your configuration, or by conditions set within the certificate, PKI Services Manager checks to be sure that the certificate has not been revoked. Depending on configuration, this check may use Certificate Revocation Lists (CRLs) or Online Certificate Status Protocol responders (OCSP).
6. If required by your application, PKI Services Manager uses the mapping rules you have configured to determine which identity or identities are allowed to authenticate using this certificate.
7. PKI Services Manager replies to the client application, letting it know if the certificate is valid and providing information about allowed identities.
8. The PKI Services Manager client application allows or denies authentication of the host or client that presented the certificate based on the information it receives from PKI Services Manager.

Configuration on Windows Systems

Start and Stop the PKI Services Manager Service on Windows

NOTE: The PKI Services Manager service starts automatically when you restart Windows.

To start the service

- From the PKI Services Manager console, click **Server** > **Start**.

  -or-

- From a DOS command window, enter the following command:

  ```
  winpki start
  ```

  -or-

- Open the Windows Services console (Control Panel >Administrative Tools > Services), select Micro Focus Reflection PKI Services Manager and click Start.

To stop the service

- From the PKI Services Manager console, click **Server** > **Stop**.

  -or-

- From a DOS command window, enter the following command:

  ```
  winpki stop
  ```

  -or-

- Open the Windows Services console (Control Panel >Administrative Tools > Services), select Micro Focus Reflection PKI Services Manager and click Stop.

To check the service status

- Start the PKI Services Manager console and look for status information on the status line at the bottom of the console window.

  -or-

- From a DOS command window, enter the following command:

  ```
  winpki ping
  ```

  -or-

- Open the Windows Services console (Control Panel >Administrative Tools > Services) and view the status of Micro Focus Reflection PKI Services Manager.

Configure PKI Services Manager on Windows

Before Reflection PKI Services Manager can validate certificates you need to customize the default configuration and map files. Use the following procedures to get started. Many additional variations are possible.

To set up your configuration and map files

1. Log in as an administrator on the computer running PKI Services Manager.
2. Start the PKI Services Manager console:

   Programs > **Micro Focus Reflection** > **Utilities** > **PKI Services Manager**

NOTE: On Windows, starting the console or the service for the first time initializes (page 11) PKI Services Manager. This creates the required data folders and default settings files. If these folders already exist, they are not changed; PKI Services Manager uses your existing data files and folders. (On UNIX the install script automatically initializes PKI Services Manager if required, and starts the service.)

3 Put a copy of the certificate (or certificates) you want to designate as a trust anchor into your certificate store (page 23). The default PKI Services Manager store is in the following location:

`ProgramData\Attachmate\ReflectionPKI\local-store\`

(This step is not required if you are using certificates in the Windows store or you have a copy of the trust anchor available somewhere else on your system.)

4 From the **Trusted Chain** pane, add your trust anchor (or anchors) to the list of trust anchors.

To use this store	Do this
Your local certificate store or a certificate file on your system	Click **Add**. Select either **Local store certificate** or **Certificate file,** click **Browse** and select the certificate for your trust anchor.
The Windows certificate store	Under **Search order to use when building path to trust anchor**, select "Windows certificate store." Click **Add**. From the Add Trust Anchor dialog box, select **Windows certificate** then click **Browse** to select an available certificate. **NOTE:** PKI Services Manager uses only those certificates that are installed for use by the local computer (not certificates installed for the current user) and are in either the trusted root certification authorities list or the trusted intermediate authorities list. To view and manage the local computer certificates, use the Microsoft Management Console. Add the Certificates Snap-in and configure it to manage certificates for the computer account.

5 From the **Revocation** pane, configure certificate revocation checking.

NOTE: By default PKI Services Manager looks for CRLs in the local store. If you use this configuration, you need to copy the CRLs to your local store.

6 From the **Identity Mapper** pane, add rules to determine which identities can authenticate with a valid certificate.

After a certificate is determined to be valid, rules are processed in order (based on rule type then sequence). If the certificate meets the requirements defined in the conditional expression (or if the rule has no condition), the allowed identities specified in that rule are allowed to authenticate. No additional rules are applied after the first match.

7 Click **File** > **Save**.

8 Start the PKI Services Manager service (page 14) if it isn't already running. If the service is already running, reload your settings (**Server** > **Reload**).

Save, Reload, and Restart on Windows

After you make changes using the PKI Services Manager console, you need to save these changes in order to update the configuration and map files.

NOTE: Saved changes do not affect subsequent certificate validation requests until you either reload your settings or restart the service.

The following settings require a restart:

Private key location
PKI server address
Enforce DOD PKI settings
FIPS mode
Maximum log files
Log output to file

All other settings changes require a reload.

To save modified settings

File > **Save**

To reload modified settings

Server > **Reload**

NOTE: Reloading the configuration also clears the internal in-memory caches used for downloading certificates and CRLs. Although certificate and CRL lifetimes are honored by the cache, it might be necessary to clear these manually if a certificate or CRL has been updated at its source before it has expired.

To restart the service

The server restarts automatically when you restart Windows, or use either of the following:

- From the PKI Services Manager console, click **Server** > **Stop**, then **Server** > **Start.**

  -or-

- From a DOS command window, enter the following command:

  ```
  winpki restart
  ```

Check Validity and Mapping on Windows

You can test whether a user or server certificate is valid and determine which identities are allowed to authenticate with that certificate. To be valid, a certificate must be signed by a trusted CA (one that is a member of a chain of trust that extends to a trust anchor that you have configured) and it must pass all other validation checks (for example, it must not be expired or revoked and all required intermediate certificates must be available).

NOTE: The certificate validation test applies only to end-entity certificates, not CA certificates. Valid CA-signed root and intermediate certificates will not pass the validation test.

To test certificates from the console

1. Start the PKI Services Manager console:

   Programs > **Micro Focus Reflection** > **Utilities** > **PKI Services Manager**

2. From the **Utility** menu, select **Test Certificate**.

3 Click **Browse.**

4 Select a certificate location, then click **Browse** to select an available certificate from that location.

5 Click **Test**.

To test certificates from the command line

1 Open a DOS command window and navigate to the program folder. The default is:

64-bit systems: `C:\Program Files (x86)\Attachmate\ReflectionPKI`

32-bit systems: `C:\Program Files\Attachmate\ReflectionPKI`

2 Use **winpki validate** to test certificates. Refer to these examples:

To	Use this command
Check if the certificate test.cer is valid.	`winpki validate \path\test.cer`
Check if the certificate is valid and if the server abc.com can authenticate with test.cer.	`winpki validate \path\test.cer -t abc.com`
Check if the certificate is valid and if the user joe can authenticate with test.cer.	`winpki validate \path\test.cer -u joe`
See which identities can authenticate with test.cer.	`winpki validate \path\test.cer -w`

Configuration on UNIX Systems

Start and Stop the Service on UNIX

The PKI Services Manager service starts automatically after installation. A script is installed, which you can use to start, stop, restart, and check the status of the service.

The following procedures use the installed **pkid** script. For additional options available using the **pkid** daemon, see PKI Services Manager Command Reference (page 55) or refer to the man page: `man pkid`

To start the service

- On Linux and Solaris: `/etc/init.d/pkid start`
- On AIX: `/etc/rc.d/init.d/pkid start`

To stop the service

- On Linux and Solaris: `/etc/init.d/pkid stop`
- On AIX: `/etc/rc.d/init.d/pkid stop`

To check the service status

- On Linux and Solaris: `/etc/init.d/pkid status`
- On AIX: `/etc/rc.d/init.d/pkid status`

Configure PKI Services Manager on UNIX

Installing the server on UNIX automatically initializes the server and starts the service, however before Reflection PKI Services Manager can validate certificates you need to customize the default configuration and map files. Use the following procedures to get started. Many additional variations are possible. For more information, see PKI Services Manager Configuration File Reference (page 58) and PKI Services Manager Map File Reference (page 62).

To set up your configuration and map files

1. Log in as root on the Reflection PKI Services Manager server.
2. Install Reflection PKI Services Manager (page 9).
3. Put a copy of the certificate (or certificates) you want to designate as a trust anchor into your certificate store (page 23). The default PKI Services Manager store is in the following location:

   ```
   /opt/attachmate/pkid/local-store
   ```

4. Open the PKI Services Manager configuration file in a text editor. The default name and location is:

   ```
   /opt/attachmate/pkid/config/pki_config
   ```

5. Use the **TrustAnchor** keyword to identify your trust anchor. For example:

   ```
   TrustAnchor = trustedca.crt
   ```

   -or-

   ```
   TrustAnchor = CN=SecureCA,O=Acme,C=US
   ```

   NOTE: To configure multiple trust anchors, add additional **TrustAnchor** lines.

6. Configure certificate revocation checking. For example:

To	Sample Configuration
Use CRLs stored on an LDAP server.	RevocationCheckOrder = crlserver CRLServers=ldap://crlserver
Use an OCSP responder.	RevocationCheckOrder = ocsp OCSPResponders = http://ocspresponder

NOTE: By default PKI Services Manager looks for CRLs in the local store. If you use this configuration, you need to copy the CRLs to your local store.

1. If intermediate certificates are required by the chain of trust in your certificates, configure access to these certificates. For example:

To	Sample Configuration
Use intermediate certificates you have added to your local store.	`CertSearchOrder=local`
Use certificates stored on an LDAP server.	CertSearchOrder=certserver CertServers=ldap://ldapserver

2. Save your changes to the configuration file.

3 Open the PKI Services Manager map file (page 62) in a text editor. The default name and location is:

```
/opt/attachmate/pkid/config/pki_mapfile
```

4 Add one or more rules to determine how the contents of a certificate determine which identities can authenticate with a valid certificate, and save your changes to the map file. For example:

```
RuleType = user
  {root joe fred susan} UPN.host Equals "acme.com"
RuleType = host
  {acme.com} Subject.CN Contains "acme"
```

For more sample rules, see Sample PKI Services Manager Mapping Rules (page 67).

NOTE: After a certificate is determined to be valid, rules are processed in order (based on rule type then sequence). If the certificate meets the requirements defined in the conditional expression (or if the rule has no condition), the allowed identities specified in that rule are allowed to authenticate. No additional rules are applied after the first match.

---

5 Test for valid PKI Services Manager configuration:

```
/usr/local/sbin/pkid -k

No errors. Configuration is valid:
```

6 Restart Reflection PKI Services Manager.

```
/usr/local/sbin/pkid restart
```

Save, Reload, and Restart on UNIX

Saving your configuration and map files does not affect subsequent certificate validation requests until you either reload your settings or restart the service.

The following settings require a restart

EnforceDODPKIMode
FipsMode
KeyFilePath
ListenAddress
LogFacility
MaxLogFiles

All other settings changes require a reload.

To reload modified settings

```
/usr/local/sbin/pkid reload
```

NOTE: Reloading the configuration also clears the internal in-memory caches used for downloading certificates and CRLs. Although certificate and CRL lifetimes are honored by the cache, it might be necessary to clear these manually if a certificate or CRL has been updated at its source before it has expired.

To restart the service

```
/etc/init.d/pkid restart
```

Check Validity and Mapping on UNIX

You can test whether a user or server certificate is valid and determine which identities are allowed to authenticate with that certificate. To be valid, a certificate must be signed by a trusted CA (one that is a member of a chain of trust that extends to a trust anchor that you have configured) and it must pass all other validation checks (for example, it must not be expired or revoked and all required intermediate certificates must be available).

NOTE: The certificate validation test applies only to end-entity certificates, not CA certificates. Valid CA-signed root and intermediate certificates will not pass the validation test.

To test certificates

- Use the **pki-val** command to test certificates. Refer to these examples:

To	Use this command
Check if the certificate test.crt is valid.	`pki-val /path/test.crt`
Check if the certificate is valid and if the server abc.com can authenticate with test.crt.	`pki-val /path/test.crt -t abc.com`
Check if the certificate is valid and if the user joe can authenticate with test.crt.	`pki-val /path/test.crt -u joe`
See which identities can authenticate with test.crt.	`pki-val /path/test.crt -w`

3 Files and Application Data

In this Chapter

- “Certificate Storage” on page 23
- “PKI Services Manager Public and Private Key” on page 24
- “PKI Services Manager Data Directories” on page 24
- “Change the Data Folder” on page 25
- “Windows Files” on page 26
- “UNIX Files” on page 27

Certificate Storage

In order to validate certificates, PKI Services Manager must have access to at least one trust anchor and may also require access to additional, intermediate certificates. One available option for storing both trust anchors and intermediate certificates is the PKI Services Manager local store. The default store location is:

Windows: *`common application data folder`*`\Attachmate\ReflectionPKI\local-store`

UNIX: `/opt/attachmate/pkid/`

You can modify this location and/or add additional stores. To do this from the console, use the **Local Store** pane. In the `pki_config` file, use the **LocalStore** keyword.

Trust Anchors

The trust anchor must be located on the computer running PKI Services Manager. PKI Services Manager can retrieve trust anchors from:

- A certificate file
- A PKCS#7 file
- (On Windows systems) The Windows Certificate Store

NOTE

- Trust anchors that are stored within a PKCS#7 file must be placed in the PKI Services Manager local store.
- Trust anchors that are stored as certificate files can be in the local store, but this is not required.
- If you configure PKI Services Manager to use the Windows store, it uses only those certificates that are installed for use by the local computer, not certificates installed for the current user. To view and manage the local computer certificates, use the Microsoft Management Console, and add the Certificates (Local Computer) Snap-in.

After your trust anchors are installed on the PKI Services Manager host, you must explicitly specify which trust anchors you want PKI Services Manager to use for certificate validation. PKI Services Manager cannot validate any certificate until the correct trust anchor for that certificate has been added to this list. To configure trust anchors from the console, use the **Trusted Chain** pane. To configure trust anchors using the `pki_config` file, use the **TrustAnchor** keyword.

Intermediate Certificates

Depending on your configuration, PKI Services Manager can retrieve intermediate certificates from one or more of the following:

- The PKI Services Manager local store
- An LDAP or HTTP server
- Any location specified in the AIA extension (page 81) of the certificate being presented
- (On Windows systems) The Windows Certificate Store

NOTE

- Certificates in the local store and in LDAP or HTTP servers can be stored as certificate files, or within a PKCS#7 file.
- PKI Services Manager can support LDAP servers that respond with more than one certificate. PKI Services Manager will determine the correct certificate to use when building a certificate path.

To configure which locations PKI Services Manager searches from the console, use **Trusted Chain** pane. In the `pki_config` file, use the **CertSearchOrder** and **CertServers** keywords.

PKI Services Manager Public and Private Key

PKI Services Manager client applications use public key authentication when connecting to PKI Services Manager to ensure the identity of the server. A public/private key pair is created automatically during PKI Services Manager initialization (page 11). Before a PKI Services Manager client application can connect to PKI Services Manager, it needs access to a copy of the PKI Services Manager public key.

The default key names are:

- Private key: `pki_key`
- Public key: `pki_key.pub`

The default key location is:

- Windows: *`data folder`*`\config\`
- UNIX: `/opt/attachmate/pkid/config/`

You can modify the key name or location. To do this from the console, go to **General** > **Private key location**. In the `pki_config` file, use the **KeyFilePath** keyword.

PKI Services Manager Data Directories

PKI Services Manager stores application data in the following base directory:

Windows:

common application data folder\Attachmate\ReflectionPKI

UNIX:
/opt/attachmate/pkid/

This directory includes the following subdirectories:

Folder	Contents
config	Configuration files (pki_config and pki_mapfile) The PKI Services Manager private and public keys
local-store	Default location for certificates and CRLs. The service does not add any files to this folder. The PKI Services Manager administrator can add certificates and CRLs to this folder and/or configure the server to search for certificates and CRLs in other locations.
logs	Log files
cache	Cached certificates and CRLs. (PKI Services Manager doesn't clear these items. If a required, cached item is removed, the server will download it again and restore the item to the cache.)
console	Directories and files used by the console. (The console maintains its own cache). The console can restore items to these locations if needed.
temp	Temporary storage used by the service. The content of this directory is cleared when the service stops.

Change the Data Folder

If you are running PKI Services Manager on Windows, you can change the PKI Services Manager data folder.

NOTE: Changing the data folder forces the service to restart. When you complete this procedure, the console closes and the service restarts automatically.

To change the data folder

1. From the **File** menu, select **Set Data Folder**.
2. Select **Use custom**.
3. Use the browse button to select a folder and click **OK**.

   NOTE: The folder must already exist, and must be on the computer running PKI Services Manager; network locations are not supported.
4. PKI Services Manager checks for an existing pki_config file in the new location.

If	Then
An existing configuration file is present	PKI Services Manager makes no change to the new location and uses the existing settings in the new location.
No configuration file is found	You can choose: **Copy existing** - PKI Services Manager copies the entire contents of your existing base directory to the new location. -or- **Create new** - PKI Services Manager generates a new default configuration file, map file, and public/private key pair in the new location.

5 Click **OK**.

NOTE: The new base directory path is saved in the Windows registry. The registry setting remains if you uninstall or upgrade the server, so subsequent installations continue to use the new location. The registry setting is created in:

HKEY_LOCAL_MACHINE\SOFTWARE\Attachmate\ReflectionPKI.

Or, on 64-bit systems:
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Attachmate\ReflectionPKI

Windows Files

Notes:

- Changes to configuration file settings do not take effect until you reload the settings or restart the service. (If a restart is required, that information is given in the keyword description.)
- By default, changes to map files do not take effect until you reload the settings or restart the service. You can modify this behavior using the map file **DynamicMapFile** setting.

The *data folder* location is configurable. The default is:
`\ProgramData\Attachmate\ReflectionPKI\`

Name	Default location	Notes
`winpki.exe`	64-bit systems: C:\Program Files (x86)\Attachmate\ReflectionPKI 32-bit systems: `C:\Program Files\Attachmate\ReflectionPKI\`	PKI Services Manager command line utility See winpki Command Reference (page 55).
`pki_config`	*data folder*`\config\`	Configuration file The console updates this file when you modify settings on any of these panes: **General**, **Local Store**, **Trusted Chain**, and **Revocation**. You can also edit the file manually. See pki_config Configuration File Reference (page 58).

Name	Default location	Notes
`pki_mapfile`	*data folder*`\config\`	Identity mapping file The console updates this file when you modify settings on the **Identity Mapper** pane. You can also edit the file manually. See "pki_mapfile Map File Reference" on page 62.
`pki_key`	*data folder*`\config\`	PKI Services Manager's private key The server uses a public/private key pair to establish its identity to calling applications.
`pki_pub`	*data folder*`\config\`	PKI Services Manager's public key Install this key on hosts running applications that make calls to PKI Services Manager.
`*.log`	*data folder*`\logs\`	PKI Services Manager log files

UNIX Files

Notes:

- Changes to configuration file settings do not take effect until you reload the settings or restart the service. (If a restart is required, that information is given in the keyword description.)
- By default, changes to map files do not take effect until you reload the settings or restart the service. You can modify this behavior using the map file **DynamicMapFile** setting.

Name	Default location	Notes
`pkid`	`/usr/local/sbin/`	PKI Services Manager daemon See PKI Services Manager Command Reference (page 55) or use: `> man pkid`

`pkid`	Linux and Solaris
	`/etc/init.d/`
	
	HP-UX
	`/sbin/init.d/`
	
	AIX
	`/etc/rc.d/init.d/`
		PKI Services Manager init script
		
		Options are **start**, **stop**, **restart**, and **status**.

`pkid-val`	`/usr/local/bin/`	Validates certificates to a running instance of Reflection PKI Services Manager.
`uninstall.sh`	`/opt/attachmate/pkid/lib/`	Uninstall script
`pki_config`	`/opt/attachmate/pkid/config/`	Configuration file See PKI Services Manager Configuration File Reference (page 58) or use: `> man pki_config`

Name	Default location	Notes
`pki_mapfile`	`/opt/attachmate/pkid/config/`	Identity mapping file See PKI Services Manager Map File Reference (page 62) or use: `> man pki_mapfile`
`pki_key`	`/opt/attachmate/pkid/config/`	PKI Services Manager's private key The server uses a public/private key pair to establish its identity to calling applications.
`pki_pub`	`/opt/attachmate/pkid/config/`	PKI Services Manager's public key Install this key on hosts running applications that make calls to PKI Services Manager.
`*.log`	`/opt/attachmate/pkid/logs/`	PKI Services Manager log files

4 PKI Services Manager Administration

In this Chapter

- "Ensuring PKI Services Manager Availability" on page 29
- "Using a Server Cluster" on page 30
- "Configure a PKI Services Manager Cluster" on page 30
- "Configure Connections via a SOCKS Proxy" on page 32
- "Changing the JRE" on page 33

Ensuring PKI Services Manager Availability

PKI Services Manager can support certificate authentication requests from multiple PKI Services Manager client applications. To help ensure that client applications have reliable access to PKI Services Manager certificate authentication services, consider the following approaches:

- Define a round-robin DNS entry for the PKI Services Manager host name, or place the PKI Services Manager host behind a load balancing server.

  NOTE: To support either of the above options, you need to use the same port and same key pair on all PKI Services Manager systems. To ensure that each of your PKI Services Manager servers returns the same validation for all certificates, make sure that all servers have identical trust anchors, configuration settings, and mapping files.

- If you are connecting from a Reflection for Secure IT server for Windows, add multiple instances of PKI Services Manager to the PKI servers list. This configuration helps ensure availability of at least one PKI server, and also balances the load among the available PKI servers.

  NOTE: To ensure that each of your PKI Services Manager servers returns the same validation for all certificates, make sure that all servers have identical trust anchors, configuration settings, and mapping files.

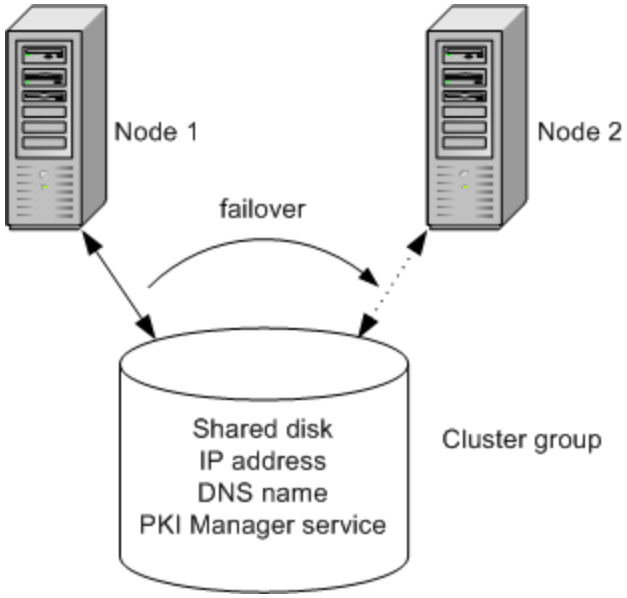
- Configure PKI Services Manager to run in a Microsoft cluster environment.

  NOTE: Although this configuration requires installing PKI Services Manager on Windows computers in a Microsoft cluster, you can use this approach to support PKI Services Manager clients running on any platform. For example, you might install PKI Services Manager in a Microsoft cluster to ensure reliable PKI server availability to Reflection for Secure IT clients and servers running on UNIX hosts.

Using a Server Cluster

You can configure PKI Services Manager to run in a Microsoft cluster environment. The Microsoft cluster service helps ensure that applications that require certificate validation services have continuous access to PKI Services Manager, even if one computer within the cluster becomes unavailable.

To run in a cluster, you install the PKI Services Manager on multiple nodes, and create a cluster group. This group defines shared resources that can be used by any node in the group. For PKI Services Manager, these shared resources include a shared disk; the PKI Services Manager IP address and DNS name; and the PKI Services Manager service. At any given time, only one node has ownership of the shared resources. If that node fails, the PKI Manager service is started on a different node and that node takes over the shared resources.



In the cluster above, if the PKI Manager service fails on Node 1, Node 2 acquires the shared resources and the service is started on the new node. At this point, Node 1 no longer has access to resources within the group. PKI Services Manager continues to run using the same configuration, so no change is apparent to clients establishing a new connection.

NOTE: Any active connections to PKI Services Manager are disconnected when a failover occurs.

Configure a PKI Services Manager Cluster

To configure a cluster, you must be running the server in a Microsoft cluster environment. The Microsoft cluster service is required to manage access to shared resources.

Install PKI Services Manager on each node of your cluster

1. Install PKI Services Manager (page 7).
2. Stop the service if it is running (page 14).

NOTE: For cluster configuration, the service should not be running until after the cluster is correctly configured.

3 Repeat this on every node that you want to include in your cluster.

Configure the cluster

1 Open the Microsoft cluster management tool (Failover Cluster Management in Windows 2008 or Cluster Administrator in Windows 2003).

2 Create a cluster group for PKI Services Manager.

3 Add the following items to the PKI Services Manager cluster group.

Resource Type	Description
Physical Disk	Location of the PKI Services Manager data folder (page 38).
IP Address	The IP address used by the server.
Network Name	The host name used by the server.

4 Add the PKI Services Manager service to the cluster group using the following settings:

Settings	Values
Resource Type	Generic Service
Generic Service Parameters	Set service name equal to: Micro Focus Reflection PKI Services Manager
	Enable this setting: **Use network name for computer name**
Dependencies	Add the following resources: Physical Disk IP Address Network Name
Registry Replication	Add this HKEY_LOCAL_MACHINE key: SOFTWARE\Attachmate\ReflectionPKI
	(If your nodes are 64-bit systems the key should be SOFTWARE\Wow6432Node\Attachmate\ReflectionPKI.)

5 Do this step only if you are running Windows 2008. It ensures that incorrect parameters are not added to the PKI Services Manager service startup command.

5a On the computer you are using to configure the cluster, open a command window as an administrator. (**Start** > **All Programs** > **Accessories**, right-click **Command Prompt** > **Run as administrator**.)

5b Enter the following command:

```
cluster res "Attachmate Reflection PKI Services Manager" /priv
```

5c If any startup parameters are configured, enter the following to clear the parameters:

```
cluster res "Attachmate Reflection PKI Services Manager" /
priv StartupParameters=""
```

5d Repeat step b to verify that there are now no startup parameters configured.

Configure PKI Services Manager

1. Open the PKI Services Manager console on the active node of your cluster group.
2. From the **File** menu, select **Set Data Folder**.
3. Select **Use custom**.
4. Set **Data folder** to a local folder on the shared physical disk you have set up as part of your cluster group, select **Enable fail-over cluster support**, and click **OK**.

   NOTE: If you have existing settings, you can elect to have these settings copied over automatically to any new location that doesn't already have PKI Services Manager settings present.
5. Configure any additional PKI Services Manager settings you want for the server.
6. Check to be sure that no files or folders configured for use by PKI Services Manager reside on any individual node in your cluster. This ensures that files accessed by users will remain available after a failover. All locally required files should be in the specified base directory. This includes the certificate store, keys, configuration file, map files, and OCSP certificates (if used).

Start PKI Services Manager

After the cluster is correctly configured, start the service:

To use	Do this
The PKI Services Manager console	Open the console on the active node and start the server (**Server** > **Start**).
The Microsoft cluster management tool	Bring the PKI Services Manager service online.

Configure Connections via a SOCKS Proxy

You can configure PKI Services Manager to connect to remote servers via a SOCKS proxy. When a SOCKS proxy is configured, all of the following connections are routed through the SOCKS proxy:

- Downloading intermediate certificates from an LDAP directory or HTTP server
- Downloading a CRL from an LDAP directory or HTTP server
- Contacting a CDP as specified in the certificate being validated
- Contacting an OCSP responder
- Contacting a server specified in AIA extension of the certificate being validated

NOTE: PKI Services Manager authenticates to the SOCKS server using the current user name (the user under which the Reflection PKI Services Manager service is running) and a blank password.

To configure a SOCKS proxy on Windows

1. Open the Windows Registry Editor and navigate to the following key (or create this key if it does not yet exist).

   64-bit systems: `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Attachmate\ReflectionPKI`

   32-bit systems: `HKEY_LOCAL_MACHINE\SOFTWARE\Attachmate\ReflectionPKI`
2. Create a string value called `JvmParams` and set the value as follows (including quotation marks):

"-DsocksProxyHost=*proxy_address* -DsocksProxyPort=*proxy_port*"

For example:

"-DsocksProxyHost=proxy.address.com -DsocksProxyPort=1080"

To configure a SOCKS proxy on UNIX

To configure a SOCKS proxy, on UNIX you need to define an environment variable called PKID_JVM_PARAMS. The basic syntax for configuring the environment variable is:

```
PKID_JVM_PARAMS = "-DsocksProxyHost=proxy_address -
DsocksProxyHost=proxy.address.com"
export PKID_JVM_PARAMS
```

NOTE: Include a single set of quotation marks around the entire variable value as shown.

To set the environment variable temporarily, you can enter the command shown above in a shell session. To create a persistent variable, you can use the following procedure.

1. Log in as root.
2. Open the `pkid` init script in a text editor. The default path is:

   Linux and Solaris:`/etc/init.d/pkid`

   HP-UX:`/sbin/init.d/pkid`
3. Under the line that reads "export PKID_HOME" add lines to define and export the new variable. For example:

   ```
   PKID_JVM_PARAMS = "-DsocksProxyHost=proxy.address.com -DsocksProxyPort=1080"
   Export PKID_JVM_PARAMS
   ```
4. Save the modified script.

Changing the JRE

PKI Services Manager installs its own Java Runtime Environment (JRE) and uses this installed JRE by default. It is also possible to configure PKI Services Manager to use a different JRE.

NOTE: The JRE you configure must be Java version 8 (1.8.0_*nn*).

Apply the Unlimited Strength Jurisdiction Policy Files to your JRE

NOTE: Each time you upgrade your JRE, you need to apply the unlimited strength policy files to the new JRE.

---

1. Download the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files from Oracle. Uncompress and extract the downloaded file.

   Be sure to download the correct policy files for your version of Java; version 8 updates(1.8.x) use a different set of files than previous versions.
2. Locate the following two policy files.

   `local_policy.jar`

`US_export_policy.jar`

3. Replace the existing limited strength policy files (located in *java-home*\lib\security on Windows or *java-home*/lib/security on UNIX) with the unlimited strength versions you extracted in the previous step.

To change the JRE on Windows

NOTE: If you upgrade PKI Services Manager, you do not need to repeat this procedure. The edited registry setting remains after an uninstall.

---

1. Open the Windows Registry Editor and navigate to the following key (or create this key if it does not yet exist).

   64-bit systems: `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Attachmate\ReflectionPKI`

   32-bit systems: `HKEY_LOCAL_MACHINE\SOFTWARE\Attachmate\ReflectionPKI`

2. Create a new string value named `JvmPath` and set the value to point to the full path where `jvm.dll` is located (*java-home*\bin\client).

   NOTE: The path to the JRE can also be set using the environment variable PKID_JVM_PATH on Windows systems. If the path is specified in both the registry and using the environment variable, the environment variable takes precedence.

To change the JRE on UNIX

To configure a JRE on UNIX you need to modify the `PKID_JVM_PATH` keyword in `/etc/pkid.conf` to point to the JRE shared library (either `libjvm.so` or `libjvm.sl` depending on your UNIX operating system), as described in the following procedure.

NOTE: If you upgrade PKI Services Manager you'll need to run `uninstall.sh` with the **upgrade** option in order to preserve your modified path setting, as described below.

---

1. Log in as root.
2. Add write permissions to `/etc/pkid.conf`:

   ```
   chmod u+w /etc/pkid.conf
   ```

3. Open `/etc/pkid.conf` in a text editor.

   Set the value of `PKID_JVM_PATH` to point to the JVM shared library. For example, on Linux:

   ```
   PKID_JVM_PATH=/usr/java/default/jre/lib/amd64/server/libjvm.so
   ```

4. Save the modified script.
5. Remove write permissions from `/etc/pkid.conf`.

   ```
   chmod u-w /etc/pkid.conf
   ```

6. Restart PKI Services Manager:

   ```
   pkid restart
   ```

To configure a separate JRE to be used only by PKI Services Manager

On some UNIX systems, if you already have a JRE on your system that you use for other purposes, you can configure a separate JRE private to PKI Services Manager. The following procedure describes how to do this on Linux systems:

1. Download the non-RPM version of the JRE.

2 Extract the JRE package.

3 Move the extracted JRE directory to a directory of your choice in the PKI Services Manager data directory (typically `/opt/attachmate/pkid`). For example:

```
mv /extracted_jvm /opt/attachmate/pkid/jre_latest
```

4 Apply the Unlimited Strength Jurisdiction Policy Files to this JRE.

5 Edit `/etc/pkid.conf` to configure PKI Services Manager to use this JRE, as described in the preceding procedure.

To preserve your modified JRE setting when upgrading on UNIX systems

This procedure creates a backup file that includes your modified path to the JRE (along with other location settings you specified when you installed PKI Services Manager). When you install the upgrade, the installer locates this backup and asks if you want to preserve your settings.

Uninstall the old version of PKI Services Manager using the upgrade option

1 Log in as root.

2 Run `uninstall.sh` using the **upgrade** option. (By default, this script is installed to `/opt/attachmate/pkid/bin/`.) For example:

```
/opt/attachmate/pkid/bin/uninstall.sh --upgrade
```

NOTE: The **upgrade** option creates a backup of your current location settings (including your modified JRE path). It does not change the default uninstall behavior for backing up the configuration directory, as described in "Upgrading From Earlier Versions" on page 10.

Install the newer version

1 Log in as root.

2 Run the install script:

```
./install.sh
```

3 If you uninstalled using the **upgrade** option, you will see a message like the following:

```
Found location settings from prior installation:
pkidHome = /opt/attachmate/pkid
pkidJvmPath = /opt/attachmate/pkid/jre_latest
systemBin = /usr/local/bin
systemSbin = /usr/local/sbin
Use locations from prior installation (y/n):
```

4 Enter `y` to preserve your settings.

5 PKI Services Manager Console

The console provides a user interface for PKI Services Manager on Windows systems. The console is not required for configuring or running PKI Services Manager. You can use the commands and configuration files described in the Reference section (page 55) of this guide on all supported systems.

In this Chapter

- “Console Menu Commands” on page 37
- “General Pane” on page 39
- “Local Store Pane” on page 41
- “Trusted Chain Pane” on page 41
- “Revocation Pane” on page 45
- “Identity Mapper Pane” on page 47

Console Menu Commands

File	
Save	Saves configuration changes. Changes are not read by the server until you reload the settings. Changes to the **General**, **Local Store**, **Trusted > Chain** and **Revocation** panes are saved to pki_config (page 82). Global changes to the **Identity Mapper** pane are saved to pki_mapfile (page 82). Certificate-specific mappings are saved to a uniquely named map file that is created in the same location.
Set Data Folder	Changes the application data folder (page 25).
Exit	Closes the console (doesn't stop the service if it is running.)
Utility	
Test certificate	Tests whether a certificate is valid and determines which identities are allowed to authenticate with a certificate.
View Public Key	Displays the fingerprint of the PKI Services Manager public key.
Server	
Start	Starts the service.
Stop	Stops the service.

Reload	Reloads changes to server configuration files without stopping the service. Changes you have saved to the configuration file do not affect the service until you reload the settings or restart the service. (Some changes require a restart. For a list of these commands, see “Save, Reload, and Restart on Windows” on page 16.) Reloading the configuration also clears the internal in-memory caches used for downloading certificates and CRLs. Although certificate and CRL lifetimes are honored by the cache, it might be necessary to clear these manually if a certificate or CRL has been updated at its source before it has expired.

Set Data Folder Dialog Box

Getting there

- From the PKI Services Manager console, go to **File** > **Set Data Folder**.

If you are running PKI Services Manager on Windows, you can change the data directory (page 24).

Use default	Use the default data folder.
Use custom	Use a custom data folder.
Data folder	Click **Browse** to specify the new data folder location. The folder must already exist, and must be on the computer running PKI Services Manager; network locations are not supported. **NOTE** ◆ If no configuration file is present in the new location, you will be given the choice of copying the contents of your existing bas directory to the new location, or creating a new, default configuration. ◆ When **Use default** is selected, the **Data folder** option is not available and any path displayed is ignored.
Enable fail-over cluster support	This option configures PKI Services Manager to run in a Microsoft cluster environment. When this option is selected, the value you specify for **Data folder** should be a local directory on the shared physical disk you have set up as part of your cluster group. **NOTE:** To configure a cluster, you must be running the server in a Microsoft cluster environment. The Microsoft cluster service is required to manage access to shared resources.

Test Certificate Dialog Box

Getting there

- From the PKI Services Manager console, go to **Utility** > **Test Certificate**.

Use this dialog box to test if a certificate is valid and to determine which identities can authenticate using a valid certificate.

NOTE: The certificate validation test applies only to end-entity certificates, not CA certificates. Valid CA-signed root and intermediate certificates will not pass the validation test.

Certificate	Click **Browse** to select the user or server certificate you want to test. You can add a certificate from your local store or the Windows certificate store. You can also specify a certificate file that's not in any store.
Operation	**Validate certificate and revocation** Validates the specified certificate and checks its revocation status. To pass a validity test, the certificate's trust anchor must be listed on the **Trusted Chain** pane. **List matched mapper rule identities** Lists the identities that can authenticate using the specified certificate base on your current **Identity > Mapper** settings. **Perform validate and mapper rule operations** Performs both of the above tests.
Results	Test results display here when you click **Test**.
Test	Click this button to test the specified certificate.

Public Key Details Dialog Box

Getting there

- From the PKI Services Manager console, go to **Utility** > **View Public Key**.

To confirm that you have correctly configured the connection between PKI Services Manager and applications that use its services, you can compare either of the public key fingerprints displayed here with values displayed for the PKI Services Manager key in those applications. The fingerprints should be identical.

Algorithm	Shows key type and size.
SHA1 fingerprint	Displays the SHA1 hash for this key (also called Bubble Babble format).
MD5 fingerprint	Displays the MD5 hash for this key.

General Pane

Getting there

- From the PKI Services Manager console, click **General**.

NOTE: You need to restart the server for some changes on this pane to take effect. For details, see the Notes section below.

The options are:

Private key location	The path to the private key used to verify the identify of Reflection PKI Services Manager. If this doesn't point to a valid key, the service won't start.

PKI server address

The address on which PKI Services Manager listens for validation requests. The default is `0.0.0.0`, which configures the server to listen on all available network adapters.

To specify a particular IP address, use the drop-down list. Available IPv4 addresses for your system are shown by default. Click "Show IPv6 addresses" to see available IPv6 addresses also.

PKI server port

The port on which PKI Services Manager listens for validation requests. The default is 18081.

Enforce DOD PKI settings

Enforces settings that meet United States Department of Defense PKI requirements.

When this option is selected, the service will not start unless the following conditions are met:

On the **General** pane:
FIPS mode is selected
Allow version 1 certificates is not selected

On the **Trusted Chain** pane:
Search order to use when building path to trust anchor does not include "Windows certificate store"

On the **Revocation** pane:
Search order to use for revocation has at least one option selected and does not include "None".

FIPS mode

Enforces security protocols and algorithms that meet FIPS 140-2 standards.

Allow version 1 certificates

Allow X.509 version 1 certificates to be used as trust anchors.

Note: Intermediate certificates must be version 3 regardless of the value of this setting.

Client debugging

Specifies whether or not debug messages are sent to the application that is requesting certificate validation.

Log output to file

Log files are created daily and saved to a directory called `logs` located in the PKI Services Manager data directory (page 82).

Maximum log files

Specifies the maximum number of log files to create. A new log file is automatically created daily. When the maximum is reached, the oldest log is removed.

Log level

Specifies the amount of information sent to the log. The log can contain both auditing messages (labeled "[audit]"), and debug messages (labeled "[debug]"). Auditing messages provide information about both successful and unsuccessful validation attempts. Debug messages are designed to help in troubleshooting.

The default log level is "Error". At this level, auditing messages are sent to the log, but debug messages are sent only if a PKI Services Manager error occurs, generally because PKI Services Manager is not correctly configured. The other options include audit messages plus increasing levels of detail in the debug messages.

Select None to turn off logging.

NOTE

- Changes made on this pane are saved to the PKI Services Manager configuration file (`pki_config`).
- Changes made on this pane do not take effect until you reload the settings (**Server** > **Reload**) or restart the server.
- Changes to the following settings require a restart: **Private key location**, **PKI server address**, **DOD PKI mode**, **FIPS mode**, **Maximum log files**, or **Log output to file**.

Local Store Pane

Getting there

- From the PKI Services Manager console, click **Local Store**.

The options are:

Local store	The local store is used to hold items that are required for certificate validation. Depending on your configuration, this may include trusted root certificates, intermediate certificates, and/or Certificate Revocation Lists (CRLs). The default local store is: *common application data folder*`\Attachmate\ReflectionPKI\local-store` You can add folders or files. When you add a folder, all the contents of the folder, including subfolders, are included in your store. Files must be binary or base 64 encoded X.509 certificates or CRLs.
Path details	Shows certificates available in the selected item under **Local Store**. To view the contents of a certificate, select it and click **View**.

NOTE

- Changes made on this pane are saved to the PKI Services Manager configuration file (`pki_config`).
- Changes made on this pane do not take effect until you reload the settings (**Server** > **Reload**) or restart the server.

Trusted Chain Pane

Getting there

- From the PKI Services Manager console, click **Trusted Chain**.

Use the **Trusted Chain** pane to determine which certificates PKI Services Manager uses to verify the authenticity of certificates presented by authenticating parties.

Trust Anchors

Trusted Anchor	Lists your trust anchors. Click **Add** to add a certificate to the list. You can add a certificate from your local store or the Windows certificate store. You can also specify a certificate file that's not in any store.
Edit	Click **Edit** to configure certificate-specific settings for revocation or identity mapping. Certificate-specific settings override the global settings configured using the **Revocation** and **Identity Mapper** panes.
Clone	Use **Clone** if you have configured certificate-specific settings and you want to add a new certificate that will use all or most of these settings. Select the certificate and click **Clone**. This displays the **Add Trust Anchor** dialog box, which you can use to add the new certificate. From the **Add Trust Anchor** dialog box, click **Properties** to view or modify the cloned settings.

Search order to use when building path to trust anchor

certificate search list	Specifies where PKI Services Manager searches for intermediate certificates. Selected locations are searched in order.

Certificate servers

Certificate servers	Lists servers from which PKI Services Manager can retrieve intermediate certificates. To add a server to the list, select "Certificate servers" under **Search order to use when building path to trust anchor**, and click **Add**. You can specify either an HTTP or an LDAP server.

NOTE

- Changes made on this pane are saved to the PKI Services Manager configuration file (`pki_config`).
- Changes made on this pane do not take effect until you reload the settings (**Server** > **Reload**) or restart the server.
- PKI Services Manager uses only those certificates that are installed for use by the local computer (not certificates installed for the current user) and are in either the trusted root certification authorities list or the trusted intermediate authorities list. To view and manage the local computer certificates, use the Microsoft Management Console. Add the Certificates Snap-in and configure it to manage certificates for the computer account.

Add Trust Anchor

Getting there

1. From the PKI Services Manager console, click **Trusted Chain**.
2. Under **Trust Anchor**, click **Add**.

Use these options to select a trust anchor:

Local store certificate	Browse for a certificate in your local store.
Windows certificate	Browse for a certificate in the Windows local computer certificate store.
Certificate file	Browse for a certificate file anywhere on your system.

Use the **Properties** button to modify settings for this trust anchor.

Properties	Click **Properties** to configure certificate-specific settings for revocation or identity mapping. Certificate-specific settings override the global settings configured using the **Revocation** and **Identity Mapper** panes.

Local Store Browser

Getting there

1. From the PKI Services Manager console, click **Trusted Chain**.
2. Under **Trust Anchors**, click **Add**.
3. Select **Local store certificate**.
4. Click **Browse**.

Use the certificate list in the **Local Store Browser** to select a certificate from your local store.

Windows Certificate Browser

Getting there

1. From the PKI Services Manager console, click **Trusted Chain**.
2. Under **Search order to use when building path to trust anchor**, select **Windows certificate store**.
3. Under **Trust Anchors**, click **Add**.
4. Select **Windows certificate**.
5. Click **Browse**.

The **Windows Certificate Browser** is available if you are running on Windows and have selected "Windows certificate store" under **Search order to use when building path to trust anchor** on the **Trusted Chain** pane.

Use the **Windows Certificate Browser** to select a certificate from the list of trusted root certification authorities in the Windows local computer certificate store.

NOTE: PKI Services Manager uses only those certificates that are installed for use by the local computer (not certificates installed for the current user) and are in either the trusted root certification authorities list or the trusted intermediate authorities list. To view and manage the local computer certificates, use the Microsoft Management Console. Add the Certificates Snap-in and configure it to manage certificates for the computer account.

Edit Trust Anchor

Getting there

1. From the PKI Services Manager console, click **Trusted Chain**.
2. Select a certificate and then click **Edit**.

Use the **Edit Trust Anchor** dialog box to configure certificate-specific settings for revocation or identity mapping. Certificate-specific settings override global settings configured using the **Revocation** (page 45) and **Identity Mapper** (page 47) panes.

Distinguished name	If you are editing properties of an existing trust anchor, this displays the certificate's Subject value. If you are configuring a new trust anchor, this is blank.
Override	Clear **Override** to configure certificate-specific values for a setting. Select **Override** to restore settings to global values.

Clone Trust Anchor

Getting there

1. From the PKI Services Manager console, click **Trusted Chain**.
2. Select a certificate and then click **Clone**.

Use the **Clone Trust Anchor** dialog box if you have configured certificate-specific settings and you want to apply all or most of these settings to a different certificate.

To clone a certificate

1. From the **Trusted Chain** pane, select a certificate and then click **Clone**.
2. Use the **Clone Trust Anchor** dialog box to add the new certificate.
3. Click **Properties** to view or modify the certificate-specific settings inherited from the original certificate.

Specify URI for Intermediate Certificate

Getting there

1. From the PKI Services Manager console, click **Trusted Chain**.
2. Under **Search order to use when building path to trust anchor**, select "Certificate servers".
3. Under **Certificate servers**, click **Add**.

Specify the **Address** value as a URI (Uniform Resource Identifier) using either LDAP or HTTP syntax. For example:

```
ldap://certserver:10389
http://certserver:1080
```

Revocation Pane

Getting there

- From the PKI Services Manager console, click **Revocation**.

The options are:

Search order to use for revocation	Determines which sources are used to check for certificate revocation and the order in which these checks occur. **NOTE:** If you select "None" and no other options are selected, no revocation checking occurs. If you select "None" along with other options, PKI Services Manager attempts to determine the revocation status using all selected options higher in the search order list. If the certificate revocation status is still unknown after these checks, authentication is allowed.
CRL servers	Lists servers from which PKI Services Manager can retrieve CRLs. To add a server to the list, select "CRL servers" under **Search order to use for revocation**, and click **Add**. You can specify either an HTTP or an LDAP server.
OCSP responder URIs	Lists OCSP responders to use for checking the certificate revocation status. To add a URI, select "OCSP responders" under **Search order to use for revocation**, and click **Add**.
OCSP certificates	Lists certificates that can be used to sign the OCSP response. This is needed only if the OCSP response does not include the signer's certificate in its response.
Settings	Opens the **Revocation Settings** dialog box, which you can use to configure policy OIDs and settings that affect how strictly revocation checking is enforced.

NOTE

- Changes made on this pane are saved to the PKI Services Manager configuration file (`pki_config`).
- Changes made on this pane do not take effect until you reload the settings (**Server** > **Reload**) or restart the server.

Specify URI for CRL Server

Getting there

1. From the PKI Services Manager console, click **Revocation**.
2. Under **Search order to use for revocation**, select "CRL servers".
3. Under **CRL servers**, click **Add**.

Specify the **Address** value as a URI (Uniform Resource Identifier) using either LDAP or HTTP syntax. For example:

```
ldap://crlserver:10389
http://crlserver:1080
```

Specify URI for OCSP Responder

Getting there

1. From the PKI Services Manager console, click **Revocation**.
2. Under **Search order to use for revocation**, select "OCSP responders".
3. Under **OCSP responder URIs**, click **Add**.

Specify the Address value as a URI (Uniform Resource Identifier) using HTTP syntax. For example:

```
http://ocsp.myhost.com
http://ocsp.myhost.com:1080
```

Add OCSP Certificate

Getting there

1. From the PKI Services Manager console, click **Revocation**.
2. Under **Search order to use for revocation**, select "OCSP responders".
3. Under **OCSP certificates**, click **Add**.

Specify a certificate that can be used to sign the OCSP response. This is needed only if the OCSP response does not include the certificate of the signer.

Revocation Settings

Getting there

- To configure global settings: **Revocation** > **Settings**
- To configure trust-specific settings: **Trusted Chain** > select a trust anchor > **Edit** > **Settings**

The options are:

Override	**NOTE:** This option is available only if you're configuring trust-specific settings. Clear **Override** to configure certificate-specific values for a setting. Select **Override** to restore settings to global values.
Policy OIDs	Enter one or more (comma-separated) OIDs to use when application policies are in force, either because **Use explicit policy** is selected or because policies are required by the certificate being presented or by a certificate within the chain of trust. Select "Any policy" to allow use of any Policy Identifier. **NOTE:** The default value is "No policy". When you select **Use explicit policy**, you must change this value to indicate which policy or policies are allowed. If **Use explicit policy** is selected and **Policy OID** is set to "No policy", no certificate can pass validation.
Use explicit policy	Select this option to enforce application policies. Use **Policy OIDs** to specify which policy or policies are allowed.
Strict validation	Specifies whether strict checking rules (as defined in RFC 3280) are used when validating certificates. Many certificates cannot pass strict checks.

Identity Mapper Pane

Getting there

- From the PKI Services Manager console, click **Identity Mapper**.

Reflection PKI Services Manager mapping binds certificates to one or more allowed identities using mapping rules. Typically, allowed identities are users or hosts. For SSH connections, to authenticate a user correctly, you need to define a rule that links information in the validated certificate to an allowed user account. The mapper provides flexible options for mapping certificates to names. You can specify allowed names explicitly in your rules, or define rules that extract information, such as user or host name, from a certificate. By using these options, you can bind identities to certificates without having to create a separate rule for each certificate. Some PKI Services Manager client applications, including Reflection Security Gateway, use PKI Services Manager for certificate validation only, and do not require any identity mapping.

NOTE

- The identity mapping requirements for PKI Services Manager clients vary. For example: The Reflection for Secure IT server supports multiple formats for specifying domain user names in map rules. The Reflection for Secure IT User Manager requires that only one user be allowed for any valid certificate. For additional information refer to information about configuring validation using Reflection for Secure IT in your product documentation.
- After a certificate is determined to be valid, rules are processed in order (based on rule type then sequence). If the certificate meets the requirements defined in the conditional expression (or if the rule has no condition), the allowed identities specified in that rule are allowed to authenticate. No additional rules are applied after the first match.
- If no true condition is found, certificate validation fails and an appropriate error message is returned to the validating application.

Rules for determining how to map a certificate to an identity

Rule

Click **Add** to configure a new rule. This opens the **Add Mapper Rule** dialog box, which you can use to construct new rules. Use the arrows to control the order in which rules are processed within each group.

To use an existing rule as a template for creating a new rule, click **Duplicate**, then select the copy and click **Edit**.

NOTE: Rules are saved to the map file, which can also be edited directly.

Rules are grouped by type. The following types are available:

user-address=*server*	The rule applies only to user certificates that are being used to authenticate to the specified server.
host	The rule applies to host certificates only.
user	The rule applies to user certificates only.
none	The rule applies to both host certificates and user certificates.

NOTE: Rule type determines the order in which rules are processed. The order for processing user certificates is: user-address, user, none. The order for processing host certificates is: host, none. Within each rule type, rules are processed in order from top to bottom.

Settings

Refresh rules from file before mapping operation	When this option is selected PKI Services Manager reloads the map file every time it evaluates a certificate to determine which identities are allowed.
Timeout for 'Extern' operations	Sets the timeout (in milliseconds) to use when you've configured an external application to handle mapping conditions. The default is 0 (zero), which sets no time out.

NOTE: Global mappings are saved to the default PKI Services Manager map file (page 82). Certificate-specific mappings are saved to a uniquely named map file that is created in the same location. Map files can be viewed and edited directly. For information about rule syntax, see PKI Services Manager Map File Reference (page 62).

Add Mapper Rule

Getting there

- From the PKI Services Manager console **Identity Mapper** pane click **Add**.

NOTE

- As you configure a rule, the constructed rule is displayed at the bottom of the dialog box. For additional information about the rule syntax see PKI Services Manager Map File Reference (page 62).
- After PKI Services Manager determines that a certificate meets the condition defined in a rule, rule processing stops.
- If the map file contains rules of multiple types, PKI Services Manager first tests user-address rules, then user rules, then the "none" rules (which apply to any certificate). PKI Services Manager stops processing rules with the first successful test.

Select the type of certificate that is to be mapped

Select the type of certificate that is to be mapped	Specifies whether the rule applies to user or host authentication. Select "Any certificate" to have the rule apply to all authentications.
Apply this rule only to this server	This option is available when the rule type is set to "User Certificate". To apply a rule only to users authenticating to a specific server, enable this setting and then specify the server. **NOTE:** When PKI Services Manager evaluates this rule, it uses the server name (not the DNS host name) of the server the user is connecting to. The server sends its name to PKI Services Manager when it requests validation of a user certificate, and PKI Services Manager uses that name when applying the rule. To determine the host name that is sent, you can enter the **hostname** command from a Windows DOS window or from a UNIX terminal session.

Specify one or more identities for the mapped certificate

Specify one or more identities for the mapped certificate

Use the text box to specify which identities can authenticate with a valid certificate. Use spaces to separate multiple allowed identities. If an allowed name includes spaces, enclose it in quotes.

For example, to allow users named root, joe, and fred smith to authenticate with a valid certificate, enter:

```
root joe "fred smith"
```

Choose certificate identity to insert

Select an item from this drop-down list to construct the allowed identity set based on the contents of the certificate presented for authentication. In the resulting rule, the percent symbol (%) precedes and follows the item you select.

For example, if you are configuring host authentication, you can select "UPN Host" to allow authentication by the host specified in the Host portion of the UPN field. The allowed identity set shows as:

```
%UPN.Host%
```

You can combine text strings with extracted information. The following example adds a Windows domain name to an extracted user identity:

```
windomain\%UPN.User%
```

NOTE: You can precede a text string with an extracted identity, and/or add a text string after an extracted identity, but you cannot combine more than one extracted value to form a single identity.

Specify how the contents of the certificate affects authentication

Accept claimed identity

When this option is selected, no conditions are set on the identity being mapped.

CAUTION: This option allows the listed identities to authenticate with any valid certificate and should therefore be used with caution.

Allow authentication if the following condition is met

When this option is selected, the set of allowed identities can authenticate only if the condition you configure is true. For details, see "Defining Conditions in a Rule" (below).

Defining Conditions in a Rule

A conditional expression takes the form:

```
Field Operation Argument
```

For *Field*, select one of the supported options from the first drop-down list.

For *Operation*, select one of the following from the second drop-down list:

Contains	Checks if the *Field* value is contained anywhere within the *Argument*.
Equals	Checks for absolute equality between the *Field* value and the *Argument* value. (This is the only option available if you select Certificate or Serial/Issuer from the first drop-down list.) For DNS, UPN and Email options, the comparison is case-insensitive.
External	Uses an external application to test the condition. Use the *Argument* box to point to the external application. Set the identity value to "First match," which is a placeholder for the value returned by the external application. PKI Services Manager sends the value of the field you specify in the first drop-down list to the external application. If the test within the external application is successful, it should exit with status 0; a non-zero return means an unsuccessful match. If you select "Certificate" in the first drop-down list, PKI Services Manager passes two arguments to your external application. The first contains the contents of the certificate in PEM format (text). The second argument contains the path to a temporary file that contains a copy of the certificate in DER format (binary). PKI Services Manager deletes the temporary DER formatted certificate when the external application exits.
Regular Expression	Applies the *Argument* as a regular expression to the *Field*. If the regular expression includes an exact match to the *Field* contents, the condition is true.

For *Argument*, enter text in the last text box. The required text depends on the Field item you have selected. For example, if you select Serial/Issuer, enter the certificate Serial number followed by the Issuer.

Fetch Certificate

Getting there

1. From the PKI Services Manager console, click **Identity Mapper**.
2. Click **Add**.
3. Select **Allow authentication if the following condition is met**.
4. From the field drop-down list, select either **Subject** or **Issuer**.
5. From the condition drop-down list, select **Equals**.
6. Click **Browse**.

Use this dialog box to locate a certificate when you are setting up a rule condition based on both serial number and certificate issuer.

Local store certificate	Browse for a certificate in your local store.
Windows certificate	Browse for a certificate in the Windows local computer certificate store.
Certificate file	Browse for a certificate file anywhere on your system.

6 Troubleshooting

In this Chapter

- “Troubleshooting PKI Services Manager Configuration” on page 51
- “Troubleshooting Identity Mapping” on page 51
- “Logging” on page 52

Troubleshooting PKI Services Manager Configuration

Use the PKI Services Manager test utility to determine if a certificate passes the validity tests. (Note: The certificate validation test applies only to end-entity certificates, not CA certificates. Valid CA-signed root and intermediate certificates will not pass the validation test.)

- “Check Validity and Mapping on Windows” on page 17
- “Check Validity and Mapping on UNIX” on page 21

Review “Certificate Attribute Requirements Enforced by PKI Services Manager” on page 78 to confirm that the certificate you are testing is valid.

If a valid certificate fails the validity test, check the following:

- Is PKI Services Manager correctly configured to point to your certificate store(s)? (In the console, check the search order on the **Trusted Chain** pane. In `pki_config`, check **CertSearchOrder**.)
- Has the required root CA been added to as a trust anchor? (In the console, check the trust anchor list on the **Trusted Chain** pane. In `pki_config`, check **Trust Anchor**.)
- Is certificate revocation correctly configured? Try turning revocation checking off to see if validation succeeds. (In the console, edit the search order on the **Revocation** pane. In `pki_config`, edit **RevocationCheckOrder**.) If you need to modify your revocation checking, review the settings on the **Revocation** pane. In `pki_config`, review **RevocationCheckOrder, CRLServers**, **OCSPCertificate**, and/or **OCSPResponders**.

Troubleshooting Identity Mapping

Problem: Updates to identity mapping don't take effect

To ensure that your settings changes take effect, save your changes (**File** > **Save**) then reload your configuration (**Server** > **Reload**). To omit the need for reloading each time, enable **Refresh rules from file before mapping operation**. If you are running on a UNIX system, use `pkid reload` after you save a modified map file, or include `DynamicFile = yes` in the map file.

Problem: Users listed as allowed identities in some rules are denied access

This problem occurs when PKI Services Manager stops processing rules before it reaches a rule that would allow access. PKI Services Manager processes rules in order from top to bottom. It stops processing rules when a certificate meets the condition defined in a rule, or if the rule has no condition defined. This means that if you include any rule with no conditions, none of the rules that come after it will ever be processed.

For example, the following configuration includes three rules with no conditions defined. In this example, the server will always stop after the first rule. The user in the first rule (joe) will always be allowed access with any valid certificate, but the other users will never be allowed access with any certificate, even if the certificate is valid.

```
{ joe }
{ don }
{ fred }
```

To allow access to multiple users without setting any rule conditions, you need to define a single rule for all users. For example:

```
{ joe don fred }
```

-or-

```
{ %UPN.User% }
```

To support processing of multiple rules, you need to include conditions in these rules. Any rule with no conditions should be at the end of the list. For example:

```
{ joe }  UPN.User Equals "joe"
{ don }  UPN.User Equals "don"
{ fred }  UPN.User Equals "fred"
{ guest }
```

Logging

PKI Services Manager logging is enabled by default. Log files are created daily and saved to a directory called `logs` located in the PKI Services Manager data directory (page 82).

You can change the logging level to control the amount of information sent to the log. The log can contain both auditing messages (labeled "[audit]"), and debug messages (labeled "[debug]"). Auditing messages provide information about both successful and unsuccessful validation attempts. Debug messages are designed to help in troubleshooting.

The default log level is "Error". At this level, auditing messages are sent to the log, but debug messages are sent only if a PKI Services Manager error occurs, generally because PKI Services Manager is not correctly configured. The additional log levels 'Warning", "Information" and "Debug" provide increasing levels of detail. ("Trace" is also available, but provides more content than is generally useful.)

NOTE: **Log level** changes don't require a restart. If you change **Maximum log files** or **Log output to file** you must restart the server.

To set the level of detail in the log file from the console (Windows)

1. From the PKI Services Manager console, go to the **General** pane.
2. Specify a value for **Log level**.
3. Save (**File** >**Save**) and reload (**Server** > **Reload**).

To change the logging level by editing pki_config (UNIX)

1. Open the PKI Services Manager configuration file in a text editor. The default name and location is:

   `/opt/attachmate/pkid/config/pki_config`

2 Use LogLevel to specify a level of detail. Allowed values are: 'error', 'warn', 'info', 'debug', and 'trace'.

3 Save the file and reload your settings (page 20).

7 Appendix

In this Chapter

- “winpki and pkid Command Reference” on page 55
- “pkid_config Configuration File Reference” on page 58
- “pki_mapfile Map File Reference” on page 62
- “Sample Mapping Rules” on page 67
- “Sample Map File with RuleType Stanzas” on page 68
- “pki-client Command Line Utility” on page 68
- “PKI Services Manager Return Codes” on page 71
- “DOD PKI Information” on page 72
- “Certificate Attribute Requirements Enforced by PKI Services Manager” on page 78

winpki and pkid Command Reference

Use **winpki** (on Windows) or **pkid** (on UNIX systems) to configure, start, and stop the PKI Services Manager service, and to check certificate validity and allowed identities.

Synopsis

Windows: `winpki [command [command args]] [options...]`

UNIX: `pkid [command [command args]] [options...]`

command = **start** | **stop** | **restart** | **reload** | **ping** | **validate <cert>**

options = [**-b** *path*] [**-c** *cert*] [**-d** *level*] [**-f** *file*] [**-h**] [**-i**] [**-k**][**-m** *path*] [**-p**] [**-o** *key=value*] [**-t** *host*] [**-u** *user*] [**-V**] [**-w**]

Commands

start

Starts the service.

stop

Stops the service.

restart

Stops and restarts the service.

reload

Reloads the configuration without stopping the service. Reloading the configuration also clears the internal in-memory caches used for downloading certificates and CRLs. Although certificate and CRL lifetimes are honored by the cache, it might be necessary to clear these manually if a certificate or CRL has been updated at its source before it has expired. Note: Most settings become available when you reload; however some settings require a restart.

ping

Displays service status and the port used by the service.

validate *certificate*

Validates a certificate and optionally provides information about allowed identities. The service must be running. For example, to determine if sample.crt is valid (UNIX syntax):

```
pkid validate sample.crt
```

Use **-u**, **-t**, or **-w** after the certificate name to get information about allowed identities for the specified certificate. For example, to determine if the user joe can authenticate using sample.cer (Windows syntax):

```
winpki validate sample.cer -u joe
```

Options

Both short (**-b** *path)* and long (**--baseDir** *path*) options are shown.

-b *path* **--baseDir** *path*

Specifies the data directory used for PKI Services Manager configuration.

-c *cert* **--cert** *cert*

Validates the specified certificate. This option is available when the service is not running. Use the **validate** command to validate certificates when the service is running.

-d *level***--debug** *level*

Specifies the amount of information sent to the log. Allowed values are: 'error', 'warn', 'info', 'debug', and 'trace'. The default is 'error'.

-f *file***--config_file** *file*

Launches using a non-default configuration file.

-h --help

Displays a brief summary of command options.

-i --init

This option is rarely needed. It initializes PKI Services Manager, which creates a key pair for the server, and creates user data directories and files. Initialization happens automatically during installation on UNIX systems and on first run on Windows systems. Using this option has no effect if your system is already initialized. Note: You can create new keys by deleting the existing keys (`pki_key` and `pki_key.pub`), and then using this option. Existing configuration files are not affected.

-k --check-config

Checks for errors in your configuration and map files and then quits.

-m *path***--migrate** *path*

Migrates certificate authentication settings from Reflection and F-Secure configuration files. If *path* specifies a directory, PKI Services Manager looks for server (`sshd2_config`) and client (`ssh2_config`) configuration files in that directory and migrates settings from those files. If *path* specifies a file, PKI Services Manager migrates the settings in the specified file. Full path information is required for both files and directories. Note: If the `pki_config` file in the destination folder already has a trust anchor configured, no migration occurs. This helps ensure that the migration won't overwrite modifications you have already configured.

Settings are migrated to the `pki_config` and `pki_map` files used by PKI Services Manager. If you use the **-b** switch, files with your migrated settings are created in the specified directory. If you omit this switch, the files are created in the default PKI Services Manager configuration directory.

A migration log is created in the `logs` directory located in the PKI Services Manager data directory. By default, this log records at a level of 'info' which shows if errors or warnings occurred. The level can be elevated using **-d**.

-o *key=value***--option** *key=value*

Sets any option that can be configured using a configuration file keyword. Options configured this way override configuration file settings. For a list of keywords and their meanings, see pki_config (page 58). Syntax alternatives are shown below. Use quotation marks to contain expressions that include spaces.

```
-o key1=value
-o key1="sample value"
-o "key1 value"
-o key=value1,value2
-o key="value1, value2"
```

To configure multiple options, use multiple **-o** switches.

```
-o key1=value -o key2=value
```

-p --showkey

Displays the public fingerprint and shows the full path and key name.

Use this option after the certificate name following a **validate** command. PKI Services Manager reads the map file(s) and reports whether the specified host is an allowed identity for the host certificate being validated.

-t *host***--hostName** *host*

Use this option after the certificate name following a validate command. PKI Services Manager reads the map file(s) and reports whether the specified host is an allowed identity for the host certificate being validated.

-u *user***--userID** *user*

Use this option after the certificate name following a **validate** command. PKI Services Manager reads the map file(s) and reports whether the specified user is an allowed identity for the user certificate being validated. If you include a server name (in the form user@server), PKI Services Manager reports on whether the user is allowed to authenticate to the specified server. If you specify only a user name, PKI Services Manager tests whether the user is allowed to authenticate with this certificate without checking for host-specific conditions.

-V --version

Displays the product name and version.

-w [*host*] **--whoAmI** [*host*]

Use this option after the certificate name following a **validate** command. PKI Services Manager reads the identity map file(s) and returns a list of all allowed identities for the certificate being authenticated. If you specify a server name after this option, the list is limited to allowed users for connections to that server. If no server name is specified, PKI Services Manager doesn't check for server-specific conditions.

pkid_config Configuration File Reference

The Reflection PKI Services Manager console saves settings to the configuration file. You can also view and edit this file manually. The default file location is:

- UNIX

  `/opt/attachmate/pkid/config/pki_config`

- Windows:

  `\ProgramData\Attachmate\ReflectionPKI\config\pki_config`

File Format

The configuration file consists of keywords followed by values. The value can be separated from the keyword by tabs, spaces, or spaces and one '='. Any line starting with a pound sign (#) is a comment. Any empty line is ignored. Some keywords can appear multiple times, and these settings are applied cumulatively. Changes to settings do not take effect until you reload the settings or restart the service. (If a restart is required, that information is given in the keyword description.)

The file includes a global section that contains settings that apply to all validation queries. You can also create stanzas that configure certificate-specific settings. The **TrustAnchor** keyword marks the beginning of each trust anchor stanza. Settings beneath the **TrustAnchor** keyword apply only to that trust anchor. The stanza ends at the next **TrustAnchor** keyword.

Some settings must be configured outside any trust anchor stanzas. These settings apply to all validation queries. Where a setting is supported both globally and within a stanza, the value within the trust anchor stanza overrides the global value.

Keywords

AllowClientStats

Specifies whether PKI Services Manager allows clients to request PKI Services Manager runtime statistics. Configure this keyword once, outside any stanza. The allowed values are 'yes' and 'no'. The default is 'yes'.

AllowVers1

Specifies whether PKI Services Manager allows version 1 certificates for a trust anchor. Note: Intermediate certificates must be version 3 regardless of the value of this setting. Configure this keyword once, outside any stanza. The allowed values are 'yes' and 'no'. The default is 'no'.

AllowWhoAmI

Specifies whether PKI Services Manager allows a client to query for the mapped identity (using **-w** or **--whoAmI**) when using PKI Services Manager to validate certificates. Configure this keyword once, outside any stanza. The allowed values are 'yes' and 'no'. The default is 'yes'.

CertSearchOrder

A comma-separated list that specifies where PKI Services Manager searches for intermediate certificates required to validate a certificate. Listed locations are searched in order. The options are 'local', 'certserver', 'aia', and 'windows'. The default is 'local, certserver.' (Note: If you select 'windows', PKI Services Manager uses only those certificates that are installed for use by the local computer, not certificates installed for the current user. To view and manage the local computer certificates, use the Microsoft Management Console. Add the Certificates Snap-in and configure it to manage certificates for the computer account.) Configure this keyword once, outside any stanza.

CertServers

Specifies a server from which PKI Services Manager can retrieve intermediate certificates when 'certserver' is included in the **CertSearchOrder** list. You can specify either an HTTP or an LDAP server. (For example: `ldap://certserver:10389` or `http://certserver:1080`) This keyword can be configured multiple times outside any stanza. The values are cumulative.

CRLServers

Specifies a server from which PKI Services Manager can retrieve Certificate Revocation Lists (CRLs) when 'crlserver' is included in the **RevocationCheckOrder** list. You can specify either an HTTP or an LDAP server. (For example: `ldap://crlserver:10389` or `http://crlserver:1080`.) This keyword can be configured multiple times outside of any stanza and multiple times per stanza. The values are cumulative.

ClientDebugging

Specifies whether the application that is requesting certificate validation can request and receive debug messages from PKI Services Manager. Configure this keyword once, outside any stanza. The allowed values are 'yes' and 'no'. The default is 'no'. Note: To view these messages you also need to set a sufficiently detailed debug level in the calling application. For the Reflection for Secure IT Server for Windows, specify "Protocol details" or higher. For the Reflection for Secure IT Client and Server for UNIX, specify debug level 3 or higher.

EnforceDODPKI

Determines whether PKI Services Manager enforces settings that meet US Department of Defense PKI requirements. The allowed values are 'yes' and 'no'. The default is 'no'. When this setting is 'yes', the service will not start unless the following conditions are met: **FipsMode** = yes; **AllowVers1** = no; **CertSearchOrder** does not include 'windows'; and **RevocationCheckOrder** has at least one option specified and does not include 'none'.

ExplicitPolicy

Determines whether PKI Services Manager enforces application policies. This keyword can be configured once outside of any stanza and once per stanza. The allowed values are 'yes' and 'no'. The default is 'no'. If the value is 'yes' you must specify one or more application policies to be enforced using the **PolicyOID** keyword. Each application policy is specified with a Policy Identifier (OID). (Note: Policies may also be required by the certificate being presented or by a certificate within the chain of trust.)

FipsMode

Enforces security protocols and algorithms that meet FIPS 140-2 standards. The allowed values are 'yes' and 'no'. The default is 'yes'. Configure this keyword once, outside any stanza. You need to restart the service if you modify this setting.

KeyFilePath

Specifies the path to the private key used to identify Reflection PKI Services Manager. When no path is specified, the path or file name is relative to the PKI Services Manager configuration directory. Configure this keyword once, outside any stanza. This setting is required. If **KeyFilePath** is not specified, or no key is present, the PKI Services Manager service will not start. The default is 'pki_key'. You need to restart the service if you modify this setting. PKI Services Manager creates a key pair when it initializes the settings, but you can also use a key pair created by **ssh-keygen** (or another tool). Only RSA keys are allowed.

ListenAddress

Specifies the port on which PKI Services Manager listens for validation requests. The syntax is `host:port`. You can specify the host name using either an IP address or a host name. IP addresses can be in either IPv4 or IPv6 format. IPv6 addresses must be enclosed in square

brackets, for example `[::D155:AB63]:18081`.The default is `0.0.0.0:18081`, which configures the server to listen on port 18081 using all available network adapters. This setting is required. You need to restart the service if you modify this setting.

LocalStore

The local store is used to hold items that are required for certificate validation. Depending on your configuration, this may include trusted root certificates, intermediate certificates, and/or Certificate Revocation Lists (CRLs). You can specify directories or files. When a directory is specified, all files in the specified directory and any subdirectories are included in the store. Files must be binary or base 64 encoded X.509 certificates or CRLs. This keyword can be configured multiple times outside any stanza. The values are cumulative. This setting is required.

LogFacility

Specifies the output location for log messages. Allowed values are 'file' and 'none'. The default is 'file'. Log files are created daily and saved to a directory called `logs` located in the PKI Services Manager data directory. Configure this keyword once, outside any stanza. You need to restart the service if you modify this setting.

LogLevel

Specifies the amount of information sent to the log. Allowed values are: 'error', 'warn', 'info', 'debug', and 'trace'. The log can contain both auditing messages (labeled "[audit]"), and debug messages (labeled "[debug]"). Auditing messages provide information about both successful and unsuccessful validation attempts. Debug messages are designed to help in troubleshooting. The default log level is 'error'. At this level, auditing messages are sent to the log, but debug messages are sent only if a PKI Services Manager error occurs, generally because PKI Services Manager is not correctly configured. The other options include audit messages plus increasing levels of detail in the debug messages. Configure this keyword once, outside any stanza.

MapFile

Specifies the location of the PKI Services Manager map file. Use the map file to configure which users or computers are allowed to authenticate with a valid certificate. When no path is specified, the path or file name is relative to the PKI Services Manager configuration directory. This setting is required. This keyword can be configured once outside of any stanza and once per stanza.

MaxLogFiles

Specifies the maximum number of log files to create. A new log file is automatically created daily. When the maximum is reached, the oldest log is removed. The default is 10. Configure this keyword once, outside any stanza. You need to restart the service if you modify this setting.

NetworkTimeout

Specifies the timeout for any network download: LDAP, HTTP, or OCSP. Units are milliseconds. The default is 20000. Configure this keyword once, outside any stanza. Configure this keyword once, outside any stanza.

OCSPCertificate

Specifies a certificate that can be used to verify the signature of the OCSP response. This is needed only if the OCSP response does not include the signer's certificate. The value can be either a certificate file or the Subject value of the certificate (for example `OcspCertificate = "CN = Secure CA, O = Secure Corporation, C = US"`). If you use the Subject value, the certificate must be in the local store. This keyword can be configured multiple times outside of any stanza and multiple times per stanza. The values are cumulative.

OCSPResponders

Specifies the address of an OCSP responder to use for checking certificate revocation when 'ocsp' is included in the **RevocationCheckOrder** list. Use an HTTP address to identify the responder. (For example: `http://ocsp.myhost.com:1080`.) This keyword can be configured multiple times outside of any stanza and multiple times per stanza. The values are cumulative.

PolicyOID

Specifies an allowed Policy Identifier (OID) to use when application policies are in force, either because **ExplicitPolicy** is 'yes' or because policies are required by the certificate being presented or by a certificate within the chain of trust. When **ExplicitPolicy** is 'yes', the specified OID must match at least one of the OIDs in the final policy set of the certificate chain. The value 2.5.29.32.0 allows use of any Policy Identifier. (Note: The default value is 'no-policy'. When **ExplicitPolicy** is set to 'yes', you must change **PolicyOID** to indicate which policy or policies are allowed; if **ExplicitPolicy** is set to 'yes' and **PolicyOID** is set to 'no-policy', no certificate can pass validation.) This keyword can be configured multiple times both outside any stanza and within a stanza. Configured values are cumulative.

RevocationCheckOrder

A comma-separated list that specifies which sources are used to check for certificate revocation and the order in which these checks occur. The options are 'ocsp', 'cdp', 'crlserver', 'local', and 'none'. The default is 'local'. Note: If you specify just 'none', no revocation checking occurs. If you specify 'none' with other options, PKI Services Manager attempts to determine the revocation status using the specified options until it reaches 'none'. If the certificate revocation status is still unknown at this point, authentication is allowed. This keyword can be configured once outside of any stanza and once per stanza.

StrictMode

Specifies whether strict checking rules (as defined in RFC 3280) are used when validating certificates. Many certificates cannot pass strict checks. The allowed values are 'yes' and 'no'. The default is 'no'. This keyword can be configured once outside of any stanza and once per stanza.

TrustAnchor

Specifies a certificate to use as the final trust point in a certificate chain of trust that Reflection for Secure IT validates. This can be an intermediate CA certificate, a root CA certificate, or a self-signed certificate (which can only validate itself). It can not be a user certificate or host certificate. The value can be either a certificate filename or the contents of the Subject field defined in the certificate (for example `TrustAnchor = "CN = Secure CA, O = Secure Corporation, C = US"`). If you specify a certificate filename and include full path information, the trust anchor is used regardless of how you configure the **CertSearchOrder** keyword. If you specify a certificate filename without including full path information, **CertSearchOrder** must include 'local'; and PKI Services Manager looks for the certificate in your local store. If you specify the contents of the certificate's Subject field, **CertSearchOrder** must include 'local' and/or 'windows'; and PKI Services Manager looks for the certificate in your local store and/or Windows certificate store. This setting is required. To configure multiple trust anchors, add additional **TrustAnchor** lines.

Note: On Windows systems, you can view the Subject value of certificates in your store using the PKI Services Manager console. On UNIX systems, you can use ssh-certview(1) to view this information.

Any keywords under a **TrustAnchor** setting create a stanza. The values you configure within a trust anchor stanza are specific to that trust anchor.

pki_mapfile Map File Reference

Reflection PKI Services Manager mapping binds certificates to one or more allowed identities using mapping rules. Typically, allowed identities are users or hosts. For SSH connections, to authenticate a user correctly, you need to define a rule that links information in the validated certificate to an allowed user account. The mapper provides flexible options for mapping certificates to names. You can specify allowed names explicitly in your rules, or define rules that extract information, such as user or host name, from a certificate. By using these options, you can bind identities to certificates without having to create a separate rule for each certificate. Some PKI Services Manager client applications, including Reflection Security Gateway, use PKI Services Manager for certificate validation only, and do not require any identity mapping.

The default map filename and location is:

- UNIX

  `/opt/attachmate/pkid/config/pki_mapfile`

- Windows:

  `\ProgramData\Attachmate\ReflectionPKI\config\pki_mapfile`

  NOTE: On Windows systems, you can modify the map file from the Reflection PKI Services Manager console using the **Identity Mapper** pane.

File Format

The map file consists of keyword settings and rules. Each rule is a single line and is independent of other rules. The format of a rule is:

{*Allowed-Identity*} [*Conditional Expression*]

After a certificate is determined to be valid, rules are processed in order (based on rule type then sequence). If the certificate meets the requirements defined in the conditional expression (or if the rule has no condition), the allowed identities specified in that rule are allowed to authenticate. No additional rules are applied after the first match.

Within the map file, you can use the **RuleType** keyword to apply different mapping criteria based on whether a user or host presents the certificate. Note: Rule type determines the order in which rules are processed. The order for processing user certificates is: user-address, user, none. The order for processing host certificates is: host, none. Within each rule type, rules are processed in order from top to bottom.

Allowed Identity Set

The allowed identity set is a required component of a rule. Allowed identities can be specified using a combination of constant values and values extracted from the certificate. The set of allowed identities can take multiple constant values, extracted values, or a combination of both. The identity mapping requirements for PKI Services Manager clients vary. For example: The Reflection for Secure IT server supports multiple formats for specifying domain user names in map rules. The Reflection for Secure IT User Manager requires that only one user be allowed for any valid certificate. For additional information refer to information about configuring validation using Reflection for Secure IT in your product documentation.

Using constant values to define allowed identities

Constant values are literal strings. Use white space to delimit separate values. (If an allowed name includes spaces, enclose it in quotes.) For example, the following rule uses literal strings to allow root, joe, and fred smith to authenticate with any valid certificate:

```
{ root joe "fred smith" }
```

NOTE: After PKI Services Manager determines that a certificate meets the condition defined in a rule, rule processing stops. In the example above, no conditions are defined. This means the rule will be applied to any valid certificate and no subsequent rules will be processed. To create a similar rule, you would need to include all allowed identities within the same rule.

Two asterisks used alone { ** } act as a wildcard for defining the allowed identity set. This option may be useful for testing, but should otherwise be used only with extreme caution. If you use this wildcard in a user rule, any user presenting a valid certificate is allowed to authenticate to any user account on the server. This creates a major security risk by allowing access to accounts with root, administrator, or power user privileges without requiring a password. If you use this wildcard in a host rule, any server with a valid certificate is accepted by the client. If you do choose to use the wildcard, consider limiting access using other options:

- Use the wildcard only with certificates signed by Certification Authorities that you control.
- Use the wildcard only in rules that have very restrictive conditions.
- Use the wildcard only in server-specific user rules (those whose **RuleType** is **user-address**).
- Limit user account access on the server side. For example, on a Secure Shell server, you might define sftp chroot jails and allow no command shell or remote command access.

Using values extracted from the certificate

Use extracted values to construct the allowed identity set based on the contents of the certificate presented for authentication. Extracted values must be preceded and followed by "%". For example, to allow authentication by the host specified in the Host portion of the UPN field:

```
{ %UPN.Host% }
```

You can also combine literal strings with extracted identities. (You can prepend a literal string to an extracted identity, and/or append a literal string, but you cannot combine more than one extracted value to form a single identity.) The following example adds a Windows domain name to an extracted user identity:

```
{ windomain\%UPN.User% }
```

Note: If the extracted identity evaluates to an empty result, the entire concatenated string is deemed to be empty and is not included in the set of allowed identities. If the entire set of allowed identities is empty, the rule is deemed to have failed and processing continues to the next rule.

Supported certificate fields are:

Subject

The Subject field defined in the certificate. The comparison is done following X.500 rules (not as a string comparison). For a successful match, the format must follow standards described in RFC 2253. To be compliant with this standard, Subject and Issuer fields start with the Common Name (for example, "CN = Secure CA, O = Secure Corporation, C = US"). On UNIX systems, you can use the **ssh-certview** utility to obtain the Subject value in this format. On Windows systems, copy the Subject contents from the Details tab of the certificate viewer, paste to an editor, and then replace new line characters with commas.

Subject.CN

The Common Name portion of the Subject field, if present.

Subject.Email

The email attribute part of the Subject, if present.

DNS

The DNS part of a SubjectAltName, if present.

IPAddress

The IP Address part of a SubjectAltName, if present. (PKI Services Manager version 1.2 and later.)

UPN

The “otherName” representation of the SubjectAltName field, with the OID of 1.3.6.1.4.1.311.20.2.3 (UPN OID), if present.

UPN.User

The userID portion of the UPN field.

UPN.Host

The host portion of the UPN field.

Email

The representation of SubjecAltName as defined in RFC 822.

Email.User

The userID portion of Email.

Email.Host

The host portion of Email.

SerialAndIssuer

The certificate serial number (hex encoded) and value of the certificate's Issuer field in this format:

serial_number Issuer

Use white space to separate the serial number from the issuer. For example:

```
461D07A8 CN = Secure CA, O = Secure Corporation, C = US
```

Cert

This indicates the entire certificate. The Operation must be **Equals** and the argument must be a file path to a certificate. Note: The Mapper does not use the certificate store defined by Reflection PKI Services Manager.

subst

This option is available when the conditional expression within a rule uses either **Regex** or **Extern**.

With **Regex**, use **subst** in combination with any regular expression that has a capturing group, which has been identified using round brackets (). If the regular expression includes an exact match to a specified certificate field, the value of the first capturing group in the expression replaces `%subst%` in the allowed identity set.

With **Extern**, use **subst** as a placeholder for the value returned by the external application.

Conditional Expression

When a conditional expression follows the {Allowed-Identity}, the allowed identities can authenticate only if the conditional expression is true. The use of a conditional expression is optional, but in most cases is recommended. If no conditional expression is included, the allowed identities can authenticate with any valid certificate.

After a certificate is determined to be valid, rules are processed in order (based on rule type then sequence). If the certificate meets the requirements defined in the conditional expression (or if the rule has no condition), the allowed identities specified in that rule are allowed to authenticate. No additional rules are applied after the first match.

The syntax for a conditional expression is:

```
Field Operation Argument
```

For *Field*, specify any of these supported certificate fields (described above): Subject, Subject.CN, Subject.Email, DNS, IPAddress, UPN, UPN.User, UPN.Host, Email, Email.Host, SerialAndIssuer, Cert, or subst.

For *Argument*, specify a string value.

For *Operation*, use one of the following:

Equals

Checks for absolute equality between the *Field* value and the *Argument* string. For DNS, UPN and Email options, the comparison is case-insensitive.

Contains

Checks if the *Field* value is contained anywhere within the *Argument* string. For DNS, UPN and Email options, the comparison is case-insensitive.

Regex

Applies the *Argument* as a regular expression to the *Field*. If the regular expression includes an exact match to the *Field* contents, the condition is true. If the set of allowed identities contains the string **%subst%**, the first capturing group (if defined) of the Regex match is inserted.

Extern

Uses an external application to test the condition. Use *Argument* to point to the application. Use `%subst%` in the allowed identity set as a placeholder for the value returned by the external application. PKI Services Manager sends the *Field* value you specify to the external application. If the test within the external application is successful, it should exit with status 0; a non-zero return means an unsuccessful match.

If the *Field* value you specify is **Cert**, PKI Services Manager passes two arguments to your external application. The first contains the contents of the certificate in PEM format (text). The second argument contains the path to a temporary file that contains a copy of the certificate in DER format (binary). PKI Services Manager deletes the temporary DER formatted certificate when the external application exits.

Sample rules with conditional expressions:

```
{ %UPN.Email% } Subject.CN Equals acme.com
```

```
{ joep } Subject Contains "Joe Plumber"
```

Rule Type Stanzas

Rule types apply different mapping criteria based on whether the validated certificate is a user certificate or a host certificate. Use the **RuleType** keyword to create a new stanza for each supported type. A stanza ends at the next **RuleType** keyword or the end of the file. The format is:

```
RuleType type
```

Valid rule types are:

none

The rule applies to both hosts and user certificates.

host

The rule applies to host certificates only.

user

The rule applies to user certificates only.

user-address= *server*

The rule applies only to user certificates authenticating to the specified server. Note: When PKI Services Manager evaluates a user-address rule, it uses the server name (not the DNS host name) of the server the user is connecting to. The server sends its name to PKI Services Manager when it requests validation of a user certificate, and PKI Services Manager uses that name when applying the user-address rule. To determine the host name that is sent, you can enter the **hostname** command from a Windows DOS window or from a UNIX terminal session.

For example, to create rules that apply only to users connecting to the server acme:

```
RuleType user-address=acme
```

Note: Rule type determines the order in which rules are processed. The order for processing user certificates is: user-address, user, none. The order for processing host certificates is: host, none. Within each rule type, rules are processed in order from top to bottom.

Keywords

DynamicFile

Specifies whether PKI Services Manager reloads the map file every time it checks for allowed identities. The allowed values are 'yes' and 'no'. The default is 'no'.

ExternTimeout

Sets the timeout for rules that use the **Extern** option. The default is 0 (zero), which sets no time out.

RuleType

Marks the beginning of a rule type stanza, which can be used to apply different mapping criteria based on whether a user or host presents the certificate. The allowed values are 'user', 'host', 'none', and 'user-address = *server*'. The default is 'none'.

Sample Mapping Rules

Rule	What happens
`{ guest }`	Because no condition is included, all valid certificates are mapped to the user "guest". This can serve as a default rule. A rule like this should go at the end of the rule list to ensure that all other rules are processed first.
`{ fred.jones } UPN.user Equals "fred"`	If the UPN representation of SubjectAltName is present, and the user part is equal to “fred”, the set of allowed identities is fred.jones.
`{ %UPN.user% } UPN.host Equals "acme.com"`	If a certificate has a UPN representation of SubjectAltName, and the host name part is "acme.com", the user name part of the UPN is returned as the set of allowed identities.
`{ guest %UPN.user% }`	If the UPN is set, the user part is included in the set of allowed identities (along with "guest"). Otherwise the set of allowed identities is "guest". Because there is no condition, this rule applies to any valid certificate.
`{ fred root } Subject.CN Contains "Fred Jones"`	If the CN of the certificate contains "Fred Jones", the set of allowed identities has two values: "fred" and "root".
`{ %subst% } Subject.CN Regex [a-zA-Z\.]*([0-9]+)`	Sets the allowed identity equal to the first numerical string within the common name portion of the Subject field. For example, if the CN is "joe.smith.12345", the allowed identity is set to "12345".
`{ elmer.foo.com } Subject.CN Contains "elmer"`	Sets the allowed identity to the fully-qualified domain name "elmer.foo.com" from a certificate that contains the short name "elmer".
`{ bob } Cert Equals /temp/certs/bob_cert.crt`	Compares the incoming certificate to the one locally stored. If they are equal, the allowed identity set is "bob".

Rule	What happens
`{ %subst% } Cert Extern /bin/myapp`	PKI Services Manager sends two values to the application "/bin/myapp". The first argument contains the contents of the certificate in PEM format (text). The second contains the path to a temporary file that contains a copy of the certificate in DER format (binary). The external application can be configured to use either of these formats. If the exit code of the called application equals 0, the allowed identity is set equal to the returned result.
`{ %UPN.User% } UPN Extern /bin/ldap-app`	In this case, an exit-code of 0 from the external application serves as confirmation that the UPN is an authorized user.
`{ %Subject.CN% %DNS% }`	Sets the allowed identity set to include the contents of either the Subject.CN field or the DNS part of SubjectAltName.
`{ windomain\%UPN.User% }`	Allows users from the specified Windows domain name to authenticate if their user name matches the UPN user name.

Sample Map File with RuleType Stanzas

```
RuleType user
# the following rules are evaluated for user certificates only:
{ scott } Subject.CN Contains acme
{ joe } Subject.CN Equals acme
{ guest }
RuleType host
# The following rule is evaluated for host certificates only:
{ elmer.acme } Subject.CN Contains elmer
RuleType user-address=myserver
# The following rule is evaluated only when myserver
# requests validation of a user certificate:
{ good %subst% } Regex UPN "([A-Za-z0-9\.-])@[*.]"
RuleType none
# "none" is the default if no RuleType is specified.
# If no rule is successfully applied from "user" or "host",
# this rule is evaluated.
{ good } SerialandIssuer contains 123 Subject.CN=foo
```

pki-client Command Line Utility

pki-client provides access to certificate validation services using Reflection PKI Services Manager.

Synopsis

`java -jar pki-client.jar validate [--service` *pki-host*[:*port*]] `--key` *public-key-file* [`--whoAmI`] [`--hostName` *host-identity*] [`--userID` *user-identity*] *certificate-file*

`java -jar pki-client.jar ping [--service` *pki-host*[:*port*]]

`java -jar pki-client.jar pubkey [--service` *pki-host*[:*port*]]

`java -jar pki-client.jar anchors [--service` *pki-host*[:*port*]]

Description

pki-client is a Java-based command line utility that you can use to query PKI Services Manager for information. You can query for information using the following keywords:

validate

Returns whether a certificate is valid, and (optionally) which servers or client users are allowed to authenticate using the certificate. Note: The certificate validation test applies only to end-entity certificates, not CA certificates. Valid CA-signed root and intermediate certificates will not pass the validation test.

ping

Returns whether the specified Reflection PKI Services Manager server is available and running.

pubkey

Returns the fingerprint of the specified Reflection PKI Services Manager server's public key in SHA1 format.

anchors

Returns the subject line of each of the trust anchors configured for the specified Reflection PKI Services Manager server.

How to run the client

To run the utility, you need a computer running Java 1.5 or newer and the `pki-client.jar` file, which is installed with Reflection PKI Services Manager. The default install location of the jar file is:

Windows:
64-bit systems: `C:\Program Files (x86)\Attachmate\ReflectionPKI`
32-bit systems: `C:\Program Files\Attachmate\ReflectionPKI\`

UNIX: `/opt/attachmate/pkid/lib/`

You can run **pki-client** on the PKI Services Manager host, or run it from a remote computer.

To configure a computer to run **pki-client**:

1. Confirm that a supported version of Java is running on the system. For example, from a command line, run the following:

   ```
   java -version
   ```

2. Copy `pki-client.jar` to any convenient location on the computer. (If you're running on the PKI Services Manager host, you can use this file in the default install location, and/or copy it to a new location.)
3. Copy the PKI Services Manager public key to the computer. (If you're running on the PKI Services Manager host, you can use the installed key file.) See the description for **--key** below for information about where to find this key.

Options

The following command line options are available.

--service *pki-host*[:*port*]

(Optional for **ping**, **pubkey**, and **anchors**) Specifies the host name or IP address of the computer running Reflection PKI Services Manager. The default is localhost:18081. You can omit this option if you're running from the PKI Services Manager host and it is configured to use the default listening address.

--key *public-key-file*

(Required for **validate**) Specifies the name and location of the public key used to confirm the identity of Reflection PKI Services Manager. Use quotation marks if the key name or path includes spaces. The default location on the PKI Services Manager host is:

UNIX: `/opt/attachmate/pkid/config/pki_key.pub`

Windows: *common application data folder*`\Attachmate\ReflectionPKI\config\pki_key.pub`

If you're running **pki-client** on a different computer than PKI Services Manager, copy the public key to the computer running **pki-client**.

--whoAmI

(Optional for **validate**) PKI Services Manager reads the identity map file(s) and returns a list of all allowed identities for the certificate being authenticated.

--hostName *host-identity*

(Optional for **validate**) PKI Services Manager reads the map file(s) and reports whether the specified host is an allowed identity for the host certificate being validated.

--userID *user-identity*

(Optional for **validate**) PKI Services Manager reads the map file(s) and reports whether the specified user is an allowed identity for the user certificate being validated.

certificate-file

(Required for **validate**) Identifies the certificate to validate. If path information is omitted, **pki-client** looks for the certificate in the current working directory. Use quotation marks if the certificate name or path includes spaces.

Examples

In all of these examples, the command line shown is executed from the same folder that contains the `pki-client.jar` file.

In the first example, **pki-client** runs on the same computer that runs the PKI Services Manager service, so no service host needs to be specified. The response indicates that the certificate is valid and that no identity checking was requested.

C:\Program Files\Attachmate\ReflectionPKI>java -jar pki-client.jar validate --key "C:\ProgramData\Attachmate\ReflectionPKI\config\pki_key.pub" c:\test\user1_sample.cer

```
Certificate is valid. Identity was not checked.
```

In the following example, **pki-client** runs on a different computer than the PKI Services Manager service, so the service host (`mypkihost`) must be specified. The public key and certificate are in the same folder as `pki-client.jar`, so no paths are required. The**--whoAmI** option is included to request a list of users who can validate with the certificate. The response indicates that only one user (joe) can authenticate using the specified certificate (`user_joe.cer`).

```
C:\Test> java -jar pki-client.jar validate --service mypkihost --key pki_key.pub
--whoAmI user_joe.cer

Certificate is valid. Allowed Identities: joe
```

The following example shows a sample response to the same command when the specified certificate failed to pass one of the required validation tests.

```
C:\Test>java -jar pki-client.jar validate --service mypkihost --key pki_key.pub
--whoAmI user_joe.cer

Certificate is not valid (error 22): Intermediate cert not found: CN=ABC Authority
```

PKI Services Manager Return Codes

Reflection PKI Services Manager returns the following codes to the application requesting validation services.

- Code 0 = No errors, successful validation.
- Codes 1-10 = Command-line errors, either with **winpki** or **pkid**.
- Codes 11-19 = Network or protocol errors.
- Codes 21-29 = Validation errors.
- Codes 31-39 = Mapper errors (certificate is valid but could not be mapped).
- Codes 41-49 = CRL or other revocation errors

Code	Meaning
0	No errors.
1	General error, unknown cause.
2	Syntax error with the command, improper arguments.
3	PKI Services Manager is already running.
4	Error in the configuration file.
5	Timeout occurred while executing the command.
6	Network error (for example, cannot connect to PKI Services Manager).
7	Access denied, user does not have permission to run the command.
8	System error. This is an internal error. Re-run with –d switch to see what happened.
9	Migration or initialization failed. See migration error log.
11	Unknown command was requested by the calling application.
12	An exception was thrown by PKI Services Manager. For more information, see the PKI Services Manager event log.
13	Syntax error with the command or packet sent to PKI Services Manager.
14	Command was ignored (not currently used, internal error).
15	Processing error. The certificate sent to PKI Services Manager is not encoded correctly.
16	Command failed (commands are: stop, reload, reconfigure).
17	Signature mismatch. Sender did not sign with a matching key.

Code	Meaning
18	Format error. The ASN protocol was not properly formatted
19	PKI Services Manager is in FIPS mode and the certificate is not valid in that mode
21	Certificate is invalid (expired, not signed, bad key, etc.)
22	No path. The issuing certificate could not be located.
23	Certificate is revoked.
24	No trust anchor. The path did not terminate to a known trust anchor.
25	Other validation error. Policy or other constraints failed.
26	Path length to the end certificate exceeded the CA path length constraint.
27	Certificate policy is invalid or does not match assertions in effect.
28	Invalid certificate signature.
29	Unknown critical extension was encountered in a certificate or CRL.
31	Identity requested did not match allowed identities.
32	No identities are allowed for this certificate (no maps exist that match).
33	Calling application did not send an identity for matching (client-side error).
34	Certificate is valid, but requested WhoAmI processing
41	Unknown CRL processing error
42	No base for a delta CRL.
43	CRL has expired.
44	Cannot verify signature or it is bad.
45	Unknown CRL extension that is marked critical.
46	Mismatch of IDP field in CRL.
47	No CRL available.

DOD PKI Information

This section describes how to install, configure, and use Reflection PKI Services Manager to operate within the Department of Defense (DOD) or other Public Key Infrastructure (PKI) environment.

Installing and Removing Trust Points

A trust point is any CA certificate in a chain of trust.

NOTE: Reflection PKI Services Manager uses only those trust points that you have explicitly configured. Certificates in other stores are not used unless you configure this.

To install and configure a trust anchor

1. Copy the certificate to the local certificate store. The default store location is:

Operating System	Default local certificate store
Windows	*common application data folder*\Attachmate\ReflectionPKI\local-store
Unix	/opt/attachmate/pkid/local-store

NOTE: You can configure other store locations. In the pki_config configuration file use the **LocalStore** keyword. Or, from the PKI Services Manager console (Windows only), go to **Local Store** > **Add**.

2 Configure PKI Services Manager to use this certificate:

Using this	Do this
Console	**Trusted Chain** >**Trust Anchors** > **Add** > **Browse** Save and reload your modified configuration.
Configuration file	Open pki_config and configure the **TrustAnchor** keyword. For example: TrustAnchor = myTrustedCA.cer

3 Save and reload your modified configuration.

To remove a trust anchor

1 Remove the certificate from your list of trust anchors.

Using this	Do this
Console	**Trusted Chain** > **Trusted Anchors** > **Remove** Save and reload your modified configuration.
Configuration file	Open pki_config and remove the **TrustAnchor** line that specifies this trust anchor, or modify it to use a different certificate.

2 Save and reload your modified configuration.

Retrieving Intermediate Certificates from an LDAP or HTTP Server

Intermediate CA trust points can be retrieved from an LDAP or HTTP server which may be identified by explicit URIs defined in the Authority Information Access (AIA) extension of a certificate, or by configuring explicit LDAP or HTTP server access using Reflection PKI Services Manager.

To configure a downloadable certificate server store using the console

1 Open the **Trusted Chain** pane.

2 In the search order list, select **Certificate servers**.

3 Under **Certificate servers**, click **Add**.

4 Specify the server using either HTTP or LDAP format. This example species an LDAP server:

```
ldap://ldapserver.myhost.com:10389
```

5 Save and reload your modified configuration.

To configure a downloadable certificate server store using the configuration file

1. Open the `pki_config` file.
2. Include 'certserver' in the **CertSearchOrder** list. For example:

   ```
   CertSearchOrder = local, certserver
   ```

3. Use **CertServers** to identify your server using either HTTP or LDAP format. This example species an LDAP server:

   ```
   CertServers = ldap://ldapserver.myhost.com:10389
   ```

4. Save and reload your modified configuration.

Configuring Certificate Revocation Checking

Revocation checking ensures that certificates used for validation have not been revoked by their issuers. Certificate revocation checking must be configured to meet DOD PKI requirements.

To configure certificate revocation checking using the console

1. Open the **Revocation** pane.

To	Do this
Use locally stored CRLs	In the search order list, select **Local store**, then copy the CRL lists to the local-store directory.
Use CRLs stored on an LDAP or HTTP server	In the search order list, select **CRL servers**. Under **CRL servers**, click **Add** and then specify the server URI.
Use an OCSP responder	In the search order list, select **OCSP**. Under **OCSP responder URIs**, click **Add** and then specify the responder URI. If your OCSP responder uses a certificate that is self-signed, or not the same as the intermediate CA certificate, you also need to specify a certificate that can be used to sign the OCSP response. Add this certificate to the **OCSP certificates** list.
Use revocation checking configured in the certificate.	In the search order list, select **CDP extension**.

2. Save and reload your modified configuration.

To configure certificate revocation checking using the configuration file

1. Open the `pki_config` file.

To	Use these example settings
Use locally stored CRLs	`RevocationCheckOrder = local` With this configuration, you need to copy the CRL lists to the local-store directory.

Use CRLs stored on an LDAP or HTTP server	`RevocationCheckOrder = certserver`
	`CRLServers = ldap://ldapserver.com`
	`-or-`
	`CRLServers = http://ldapserver.com`

Configure an OCSP responder when no OCSP responder is configured in the certificate's AIA extension	`RevocationCheckOrder = ocsp`
	`OCSPResponders = http://ocsp.myhost.com`
	If your OCSP responder uses a certificate that is self-signed, or not the same as the intermediate CA certificate, you also need to specify a certificate that can be used to sign the OCSP response. Add this certificate to the **OCSP certificates** list.

Use an OCSP responder configured in the certificate's AIA extension.	`RevocationCheckOrder = ocsp`
	Include 'aia' in the certificate search order. For example:
	`CertSearchOrder = local, aia`

Use revocation checking configured in the certificate.	`RevocationCheckOrder = cdp`

2 Save and reload your modified configuration.

Configuring PKI Services Manager to Meet DOD Requirements

By default, PKI Services Manager allows some configurations that do not meet DOD PKI requirements. To ensure that certificate validation meets DOD requirements, refer to the following procedures.

To configure DOD requirements using the console

1 Install and configure at least one trust anchor.

2 From the **General** pane:

- Select **Enforce DOD PKI Settings**.
- Select **FIPS Mode**.
- Clear **Allow version 1 certificates**.

3 From the **Trusted Chain** pane:

- Under **Search order when building path to trust anchor**, ensure that "Windows certificate store" is *not* selected.

4 From the **Revocation** pane:

- Under **Search order to use for revocation**, ensure that "None" is *not* selected.
- Select and configure at least one option for checking certificate revocation.

5 Save your settings and restart the service.

To configure DOD requirements using the configuration file

1. Install and configure at least one trust anchor.
2. Open the `pki_config` file.
3. Configure the following:

```
EnforceDODPKI = yes
FipsMode = yes
AllowVers1 = no
```

4. Use **RevocationCheckOrder** ensure that "none" is *not* included in the list of options, and configure at least one option for checking certificate revocation.
5. Ensure that "windows" is *not* included in the list of options specified for **CertSearchOrder**.
6. Save your settings and restart the service.

Configuring Micro Focus Products to Use PKI Services Manager for Certificate Authentication

After Reflection PKI Services Manager is correctly configured, you must also configure the Reflection products that use PKI Services Manager for certificate authentication. For details, search on "PKI Services Manager" in the product documentation.

Private Key Safeguards

If a client private key is stolen, a malicious user can gain access to files on any servers accessible to that user. If a server private key is stolen, a malicious user can use this key to accomplish an impersonation attack, in which another server poses as your host. Use the following guidelines to minimize these risks.

Protecting private keys on the client:

- Each client user should always protect his or her private key with a passphrase. This ensures that only someone who knows the passphrase can authenticate with that key.
- Users should create and protect passphrases following your the specifications for password length and complexity in your organization’s Security Policy.
- File permissions on the private key should be set so that only the user has access to the key.

Protecting private keys on the server:

- Micro Focus servers enforce permissions on server private keys to ensure that only the server administrator has access to private keys. If key permissions are altered to allow greater access in a way that allows other access, the server resets correct permissions and logs a warning. If you see this warning, you should investigate to determine the cause.

Actions to Take if a Key is Compromised

Consider a private key compromised if it has become available to any unauthorized entity, or if you have reason to distrust the actions of any person who has access to the key.

If a private key is compromised, revoke the client certificate.

To replace a compromised key:

1. Obtain a new private key and certificate
2. Replace the compromised key, and update the PKI Services Manager client application to authenticate using the new key.

To remove the compromised key

1. Remove the key from the local store using a DOD-approved file erasure utility.
2. If the original file containing the old key and certificate (*.pfx or *.p12) is still on the client computer, use a DOD-approved file erasure utility to delete this file.

Using Uniform Resource Identifiers for DOD PKI Services

PKI Services Manager supports the use of URIs for automatic retrieval of updated CRL lists as defined in section 4.2.1.14 of RFC3280.

PKI Services Manager checks for certificate revocation as follows:

1. Check the `crl_cache` for valid revocation information. If none is found, continue on to step 2.
2. If CDP checking is enabled, check the CDP extension in the certificate for HTTP or LDAP URIs and query these in the order specified (first HTTP, then LDAP). If the certificate is found to be revoked, the validation fails. If the certificate is not found continue on to step 3.
3. If download from a CRL server is enabled and one or more CRL servers are configured for PKI Services Manager, assemble the Distinguished Name for the CA listed in the Issuer extension of the certificate and query for the CRL file. If the certificate is not found to be revoked in any CRL, continue to the next validation step.

Updates for expired CRLs are handled automatically, and do not require administrator intervention or configuration.

If OCSP checking is enabled, PKI Services Manager always checks all available OCSP responders to ensure that the connection will fail if any of these responders knows that the certificate has been revoked. For the connection to succeed at least one OCSP responder must be available and return a value of 'good' for the certificate status. PKI Services Manager performs these checks as follows.

1. If AIA extension checking is enabled, check the AIA extension in the certificate for one or more OCSP responders and query each of those responders. If the status of the certificate comes back as 'revoked' from any responder, the validation fails.
2. Check for one or more user-configured OCSP responders and query each of those responders. If the status of the certificate comes back as 'revoked' from any responder, the validation fails.
3. If all responders returned 'unknown' the validation fails. If a 'good' response was returned from at least one of the queried OCSP responders continue on to the next validation step.

Using URIs to Retrieve Intermediate Certificates

As defined in section 4.2.2.1 of RFC3280, PKI Services Manager can use URIs to retrieve intermediate CA certificates as follows:

1. If the local store is enabled, check the cert_cache file for the required intermediate certificate. If it is not found, continue on to step 2.
2. If AIA is enabled, and either HTTP or LDAP URIs are defined in the Authority Information Access (AIA) extension of a certificate, attempt to use these (first HTTP, then LDAP) to retrieve intermediate CA certificates.
3. If download from a certificate server is enabled, and one or more servers are configured in the certificate servers list, the preceding attempts fail, assemble a Distinguished Name from the issuing certificate's Subject Name, and queries the defined LDAP or HTTP server for the contents of the CACertificate attribute.

Certificate Attribute Requirements Enforced by PKI Services Manager

This topic provides a detailed list of which certificate fields are checked by PKI Services Manager, and what requirements must be met for a certificate to be accepted as valid.

- Requirements for All Certificates (page 78)
- Requirements for CA Certificates (page 78)
- Requirements for SSL, TLS, and FTPS Server Certificates (page 79)
- Requirements for Secure Shell (SSH) and SFTP Server Certificates (page 79)
- Requirements for User Certificates (page 80)

Requirements for All Certificates

The following version 1 fields MUST all contain valid data.

Field	Validation information for this field and its attributes
Version	Version 3 is required for user or server certificates. The version accepted for CA certificates is configurable (on the **General** pane or using the **AllowVers1** keyword), but by default version 1 certificates are rejected.
Serial number	Used in combination with Issuer to identify this certificate for revocation checking.
Issuer	Used to build the chain of trust for this certificate. -and- Used in combination with Serial number to identify this certificate for revocation checking.
Subject	The CN attribute is used to determine the identity of the entity presenting this certificate. (Note: In some certificates, the Subject Alternate Name extension is used as an alternate method of specifying identity.)
Valid from Valid to	Used to determine if the certificate is within the valid time period.
Signature algorithm Signature hash algorithm	Provides information required to decrypt the certificate's signature.
Public key	Used to decrypt the digital signatures provided by the certificate owner.

Requirements for CA certificates

Certificate Authority (CA) certificates must meet the following version 3 extension requirements in addition to the version 1 requirements listed in Requirements for All Certificates (page 78).

Field	Validation information for this field and its attributes
Basic Constraints	MUST be set as a critical extension. Subject type MUST be set to CA. Path Length Constraint is not required. If present, it will be used to check the length of the chain.
Key Usage	MUST be present. May be set as a critical extension. MUST include Certificate signing. May also include CRL signing, Off-line CRL Signing, Digital Signature. (These attributes may be required if the CA server also issues CRLs or OCSP responses.)
Authority Information Access	Not required. If present, it can be used to retrieve the issuer certificate and/or determine OCSP responder servers.
CRL Distribution Points	Not required. If present, it can be used to retrieve CRLs.
Certificate Policies	Not required. May contain one or more policy OIDs, which, if present, must also be present in the Certificate Policies field of other certificates up and down the chain of trust.

Requirements for SSL, TLS, and FTPS Server Certificates

Certificates used to authenticate SSL, TLS, and FTPS servers must meet the following version 3 extension requirements in addition to the version 1 requirements listed in Requirements for All Certificates (page 78).

Field	Validation information for this field and its attributes
Key Usage	May be present, but not required. If present: MUST include Digital Signature and Key Encipherment. May also include Non Repudiation, Data Encipherment and others, but these are ignored.
Extended Key Usage (Enhanced Key Usage is an equivalent name.)	May be present, but not required. If present: MUST include Server authentication.
Authority Information Access	Not required. If present, it can be used to retrieve the issuer certificate and/or determine OCSP responder servers.
CRL Distribution Points	Not required. If present, it can be used to retrieve CRLs.
Certificate Policies	Not required. May contain one or more policy OIDs, which, if present, must also be present in the Certificate Policies field of other certificates up the chain of trust.
Subject Alternative Name	Not required. May be used to determine alternate names for the server presenting the certificate using either the dNSName or iPAddress attributes.

Requirements for Secure Shell (SSH) and SFTP Server Certificates

Certificates used to authenticate Secure Shell (SSH) and SFTP servers must meet the following version 3 extension requirements in addition to the version 1 requirements listed in Requirements for All Certificates (page 78).

Field	Validation information for this field and its attributes
Key Usage	May be present, but not required. If present: MUST include Digital Signature and Key Encipherment. May also include Non Repudiation, Data Encipherment and others, but these are ignored.
Extended Key Usage (Enhanced Key Usage is an equivalent name.)	May be present, but not required. If present: MUST include Server authentication.
Authority Information Access	Not required. If present, it can be used to retrieve the issuer certificate and/or determine OCSP responder servers.
CRL Distribution Points	Not required. If present, it can be used to retrieve CRLs.
Certificate Policies	Not required. May contain one or more policy OIDs, which, if present, must also be present in the Certificate Policies field of other certificates up the chain of trust.
Subject Alternative Name	Not required. May be used to determine alternate names for the server presenting the certificate using either the dNSName or iPAddress attributes.

Requirements for User Certificates

Certificates used to authenticate client users must meet the following version 3 extension requirements in addition to the version 1 requirements listed in Requirements for All Certificates (page 78).

Field	Validation information for this field and its attributes
Key Usage	May be present, but not required. If present: MUST include Digital Signature and Key Encipherment. May also include Non Repudiation, Data Encipherment and others, but these are ignored.
Extended Key Usage (Enhanced Key Usage is an equivalent name.)	May be present, but not required. If present: MUST include Client authentication.
Authority Information Access	Not required. If present, it can be used to retrieve the issuer certificate and/or determine OCSP responder servers.
CRL Distribution Points	Not required. If present, it can be used to retrieve CRLs.
Certificate Policies	Not required. May contain one or more policy OIDs, which, if present, must also be present in the Certificate Policies field of other certificates up the chain of trust.
Subject Alternative Name	Not required. May be used to determine alternate names for the user presenting the certificate using the rfc822Name or otherName attributes.

Glossary of Terms

AIA (Authority Information Access). The AIA extension is a field within a certificate that contains a Uniform Resource Identifier (URI) used to locate an item required to validate this certificate. The URI can point to an OCSP responder or to a certificate from the issuing Certificate Authority (CA).

authentication. The process of reliably determining the identity of a communicating party. Identity can be proven by something you know (such as a password), something you have (such as a private key or token), or something intrinsic about you (such as a fingerprint).

CA (Certificate Authority). A server, in a trusted organization, which issues digital certificates. The CA manages the issuance of new certificates and revokes certificates that are no longer valid for authentication. A CA may also delegate certificate issuance authority to one or more intermediate CAs creating a chain of trust. The highest level CA certificate is referred to as the trusted root.

CDP (CRL Distribution Point). A CDP is the location where you can download the latest CRL. A CDP is typically listed in the CRL Distribution Points field of the Details tab of the certificate.

CRL (Certificate Revocation List). A digitally signed list of certificates that have been revoked by the Certification Authority. Certificates identified in a CRL are no longer valid.

digital certificate. An integral part of a PKI (Public Key Infrastructure). Digital certificates (also called X.509 certificates) are issued by a certificate authority (CA), which ensures the validity of the information in the certificate. Each certificate contains identifying information about the certificate owner, a copy of the certificate owner's public key (used for encrypting and decrypting messages and digital signatures), and a digital signature (generated by the CA based on the certificate contents). The digital signature is used by a recipient to verify that the certificate has not been tampered with and can be trusted.

digital signature. Used to confirm the authenticity and integrity of a transmitted message. Typically, the sender holds the private key of a public/private key pair and the recipient holds the public key. To create the signature, the sender computes a hash from the message, and then encrypts this value with its private key. The recipient decrypts the signature using the sender's public key, and independently computes the hash of the received message. If the decrypted and calculated values match, the recipient trusts that the sender holds the private key, and that the message has not been altered in transit.

hash. A hash (hash value or message digest) is a fixed-length number generated from variable-length digital data. The hash is substantially smaller than the original data, and is generated by a formula in such a way that it is statistically unlikely that other data will produce the same hash value.

OCSP (Online Certificate Status Protocol). A protocol (using the HTTP transport) that can be used as an alternative to CRL checking to confirm whether a certificate is valid. An OCSP responder responds to certificate status requests with one of three digitally signed responses: "good", "revoked", and "unknown". Using OCSP removes the need for servers and/or clients to retrieve and sort through large CRLs.

PKCS. PKCS (Public Key Cryptography Standards) is a set of standards devised and published by RSA laboratories that enable compatibility among public key cryptography implementations. Different PKCS standards identify specifications for particular cryptographic uses. PKI Services Manager uses the following PKCS standards:

PKCS#7 can be used to sign and/or encrypt messages. It can also be used to store certificates and to disseminate certificates. PKI Services Manager can use certificates stored in this format.

PKCS#10 is used for certificate requests to a Certificate Authority (CA).

PKI Services Manager Configuration File. Configuration settings are saved to `pki_config`. The default location is:

UNIX: `/opt/attachmate/pkid/config/pki_config`

Windows: `\ProgramData\Attachmate\ReflectionPKI\config\pki_config`

PKI Services Manager Configuration Directory. The configuration directory stores the `pki_config` and `pki_map` configuration files. It also contains the public/private key pair to establish its identity to calling applications. The default location is:

UNIX: `/opt/attachmate/pkid/config/`

Windows: `\ProgramData\Attachmate\ReflectionPKI\config\`

PKI Services Manager Data Directory. The user data directory is configurable. The default is:

UNIX: `/opt/attachmate/pkid`

Windows: `\ProgramData\Attachmate\ReflectionPKI\`

PKI Services Manager Map File. The map file binds certificates to one or more allowed identities using mapping rules.

UNIX: `/opt/attachmate/pkid/config/pki_mapfile`

Windows: `\ProgramData\Attachmate\ReflectionPKI\config\pki_mapfile`

public key/private key. Public keys and private keys are pairs of cryptographic keys that are used to encrypt or decrypt data. Data encrypted with the public key can only be decrypted with the private key; and data encrypted with the private key can only be decrypted with the public key.

regular expression. Often abbreviated as *regex*, a regular expression is a string of characters that describes one or more matching strings. Within a regular expression, some characters have a predefined meaning that determines what qualifies as a match. For example, the regular expression "t.*t" matches any word that starts and ends in the letter *t*, while the regular expression "text" matches only itself.

root CA certificate. A certificate created and signed by a trusted certification authority that is the final trust point in a certificate chain of trust. In a trusted root CA, the certificate's Issuer is the same as the certificate's Subject, and, in the Basic Constraints field, the Subject type must be set to CA.

self-signed certificate. A certificate that was created and signed by an end-entity (usually a server) where the Issuer equals the Subject. When this is created by a server, the CA bit in the Basic Constraints is not set, and this certificate can only vouch for itself; it cannot sign other certificates.

trust anchor. A certificate that can be used as the final trust point in a certificate chain of trust. Note: PKI Services Manager validates certificates using only those trust anchors that have been explicitly configured for use by PKI Services Manager. You can configure a trust anchor using a root CA certificate, an intermediate CA certificate, or a self-signed certificate (one that can only validate itself).

URI (Uniform Resource Identifier). A string of characters that represents the location or address of a resource. URIs can be used to locate resources on the Internet or on an LDAP server.

Windows common application data folder. The application data folder is hidden by default.

The default is: `\ProgramData\`