Symantec™ PKI Client

Writing Post-processing Scripts Guide
2.11.0

Symantec

Legal Notice

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

http://www.symantec.com

Contents

Chapter 1

Writing Post Processing Scripts

This chapter includes the following topics:

About Writing Post Processing Scripts

Once PKI Client has completed operations on a certificate, your application must be able to consume them. You can write BAT (batch) for Windows or SH (shell) script files for Mac to notify your applications of the certificate status and location. Your application can then use these certificates.

Before PKI Client calls the script file, it writes certificate data to files that is made accessible to the script files. The script files use this data to perform the requested operation. The fully-qualified path to the certificate files is provided to the script file. After post processing is complete, the certificate file is deleted.

This documentation describes how to write script files to notify your applications of certificate status, location, and availability after the following certificate operations:

- Enroll (new certificate)

- Import (existing certificate)
- Renew (new certificate, old certificate)
- Smart Card Insert (certificate on security device)
- Diagnostic Mode

Note: PKI Client allows users to import expired certificates. For example, importing expired certificates is sometimes useful, to allow users to access email previously encrypted by an expired certificate. To verify the validity of a certificate before importing it, use the Certificate Property utility.

See "About the Certificate Properties Utility" on page 13.

Once you prepare the script files, you upload them to PKI Manager and assign them to one or more certificate profiles from the **Manage Profiles** page. You can also remove them from a specific certificate profile on the **Manage this profile** page for that certificate profile. Once assigned, the script files then run for every certificate that has been issued from that profile.

See PKI Manager and its associated help for procedures on how to upload the script files and assign them to certificate profiles.

BAT File Format

Windows BAT files must end with a blank line and begin with the following line:

```
@ECHO OFF
```

SH File Format

Mac SH files must end with a blank line and begin with the appropriate hash-bang (#!), usually:

```
#!/bin/sh
```

Additionally, all error codes must be between 1 and 255. The code 195 is reserved and should not be used.

About Script File Parameters

PKI Client passes the following information to your script file:

- The operation, as mentioned in the operations section.

- The path to the PKI Client install directory (necessary if the script file needs to locate other dependencies, such as binaries).
- The path to the new or the imported certificate file, as appropriate.
- The path to the old certificate file (for renewal only; otherwise this parameter is not present).
- The path to the root certificate of the issuing CA.
- The paths to any intermediate certificates between the root and the user certificate.

Special Considerations

When you create the script files, consider the following issues:

- If your script file depends on or calls other applications, these applications must be present in the `<PKI_Client_install_dir>\BERETTA\BIN` directory.
- Your script file can invoke a command line utility obtain information about a specific certificate.
  See "About the Certificate Properties Utility" on page 13.
- Managed PKI does not control the order in which script files are run.
  If you use multiple script files to perform operations on a single certificate, they should not cause any side-effects that affect any other script files. You must make sure that all script files do not rely on output of other script files.
  If you want to control the order of certificate operations, you need to merge all certificate operations into a single script file. You can use the single script file to control the dependencies between operations.

About Sample BAT Files

The following sample BAT file writes example output to the screen, although it does not install a certificate.

PKI Client executes your script file and passes the following parameters to your script file in the following order:

- Operation (Enroll, Import, or Renew)
- PKI Client installation directory
- The user's certificate that was enrolled, imported, or renewed
- The user's expiring certificate (only applies to Renew operations)
- The root CA certificate

- The intermediate CA certificates

Note: For successful operations you must end with an exit code of 0. For any error conditions, you must end with an exit code of 1000 or higher.

Non-zero exit codes displayed to your end users are also written to the `beretta.log` file. Use a different error code for each type of error to aid in troubleshooting issues.

```
@ECHO OFF

if "%1" == "Enroll" (GOTO enroll)
if "%1" == "Import" (GOTO import)
if "%1" == "Renew" (GOTO renew)

ECHO Operation not supported: %1
EXIT /B 1

:enroll
ECHO Adding new certificate
ECHO The PKI Directory is %2
ECHO The New Certificate is %3
ECHO The Root CA Certificate is %4
ECHO The Intermediate Certificates are %5, %6, %7, %8, %9
EXIT /B 0

:import
ECHO Adding imported certificate
ECHO The PKI Directory is %2
ECHO The Imported Certificate is %3
ECHO The Root CA Certificate is %4
ECHO The Intermediate Certificates are %5, %6, %7, %8, %9
EXIT /B 0

:renew
ECHO Renewing certificate
ECHO The PKI Directory is %2
ECHO The New Certificate is %3
ECHO The Old Certificate is %4
ECHO The Root CA Certificate is %5
ECHO The Intermediate Certificates are %6, %7, %8, %9
EXIT /B 0
```

Sample BAT File Executions

```
*Enroll:
Sample.bat Enroll "C:\Program Files\Symantec\PKI Client\" "C:\
Somepath\New.cer" C:\Somepath\Root.cer" "C:\Somepath
\Intermediate1.cer" "C:\Somepath\Intermediate2.cer"
Output:
Adding new certificate
The PKI Client Directory is "C:\Program Files\Symantec\PKI Client\"
The New Certificate is "C:\Somepath\New.cer"
The Root CA Certificate is "C:\Somepath\Root.cer"
The Intermediate Certificates are "C:\Somepath\Intermediate1.cer",
"C:\Somepath\Intermediate2.cer", , ,

*Import:
Sample.bat Import "C:\Program Files\Symantec\PKI Client\" "C:\
Somepath\Import.cer" C:\Somepath\Root.cer" "C:\Somepath
\Intermediate1.cer" "C:\Somepath\Intermediate2.cer"
Output:
Adding imported certificate
The PKI Client Directory is "C:\Program Files\Symantec\PKI Client\"
The Imported Certificate is "C:\Somepath\Import.cer"
The Root CA Certificate is "C:\Somepath\Root.cer"
The Intermediate Certificates are "C:\Somepath\Intermediate1.cer",
"C:\Somepath\Intermediate2.cer", , ,

*Renew:
Sample.bat Renew "C:\Program Files\Symantec\PKI Client\" "C:\
Somepath\New.cer" C:\Somepath\Old.cer" "C:\Somepath
\Root.cer" "C:\Somepath\Intermediate1.cer"
"C:\Somepath\Intermediate2.cer"
Output:
Renewing certificate
The PKI Client Directory is "C:\Program Files\Symantec\PKI Client\"
The New Certificate is "C:\Somepath\New.cer"
The New Certificate is "C:\Somepath\Old.cer"
The Root CA Certificate is "C:\Somepath\Root.cer"
The Intermediate Certificates are "C:\Somepath\Intermediate1.cer",
"C:\Somepath\Intermediate2.cer", ,

*Other:
Sample.bat Other "C:\Program Files\Symantec\PKI Client\" "C:\Somepath\
New.cer" C:\Somepath\Root.cer" "C:\Somepath\
```

```
Intermediate1.cer" "C:\Somepath\Intermediate2.cer"
Output:
Operation not supported: Other
```

Sample SH File

The following sample SH file writes example output to the screen, although it does not install a certificate.

PKI Client executes your script file and then passes the following parameters to your script file in the following order:

- Operation (Enroll, Import, or Renew)
- PKI Client installation directory
- The user's certificate that was enrolled, imported, or renewed
- The user's expiring certificate (only applies to Renew operations)
- The root CA certificate
- The intermediate CA certificates

Note: For successful operations you must end with an exit code of 0. For any error conditions, you must end with an exit code of 1000 or higher.

Non-zero exit codes displayed to your end users are also written to the `beretta.log` file. Use a different error code for each type of error to aid in troubleshooting issues.

```
#!/bin/sh

if [ "$1" == "Enroll" ]
    echo Adding new certificate
    echo The PKI Directory is $2
    echo The New Certificate is $3
    echo The Root CA Certificate is $4
    echo The Intermediate Certificates are $5, $6, $7, $8, $9

elif [ "$1" == "Import" ]
    echo Adding imported certificate
    echo The PKI Directory is $2
    echo The Imported Certificate is $3
    echo The Root CA Certificate is $4
    echo The Intermediate Certificates are $5, $6, $7, $8, $9
    exit 0
```

```
elif [ "$1" == "Renew" ]
    echo Renewing certificate
    echo The PKI Directory is $2
    echo The New Certificate is $3
    echo The Old Certificate is $4
    echo The Root CA Certificate is $5
    echo The Intermediate Certificates are $5, $6, $7, $8
    exit 0

else
    echo Operation not supported
    exit 1
fi

exit 0
```

Sample SH File Executions

```
S*Enroll:

./Sample.sh Enroll "/usr/local/lib/tblive-4" "/Somepath/New.cer"
"/Somepath/Root.cer" "/ Somepath/Intermediate1.cer"
"/Somepath/Intermediate2.cer"

Output:

Adding new certificate

The PKI Client Directory is "/usr/local/lib/tblive-4" The New
Certificate is "/Somepath/New.cer" The Root CA Certificate is
"/Somepath/Root.cer" The Intermediate Certificates are
"/Somepath/Intermediate1.cer", "/Somepath/Intermediate2.cer", , ,
```

Modifying Template Files for Specific Implementations

Symantec provides template versions of the script files that you can modify for your specific implementations.

To modify template files for specific implementations

1. Download a template script file from the **Create custom scripts page** of PKI Manager.
2. Modify the template file to meet your needs.

   Typically you should only need to modify entries between the `START CONFIG` and `END CONFIG` headers.

   See "Template File Configuration Values" on page 11.
3. Upload the modified file to your account from the **Create custom script page**.
4. Assign the newly uploaded file to the appropriate certificate profile on the **Manage certificate profile** page.

Template File Configuration Values

Table 1-1 describes the values that you can modify in the script template files.

Table 1-1 Templates and Configuration Values

Template	Usage	Values to Modify
ADPub.bat **Note:** For the certificates that are enrolled using PKI Enterprise Gateway or the Autoenrollment Server, use profile creation. This functionality is configured under **Advanced certificate options > Publish to company directory**.	Publish the certificate to an Active Directory. This specific script template file is supported on Windows only for the certificates that are stored at Symantec. (For certificates that are enrolled using PKI Enterprise Gateway or the Autoenrollment Server, use profile creation).	No configuration is necessary to use this template.
Cisco.bat	Configure a certificate to connect to a Cisco router. Supported on Windows only	■ ProfileName - The name of the Cisco batch file. ■ Host - IP address of the Cisco VPN gateway host.

Table 1-1 Templates and Configuration Values *(continued)*

Template	Usage	Values to Modify
Juniper.bat and Juniper.sh	Configure a certificate to connect to a Juniper router. Supported on Windows (.bat) or Mac OSX (.sh)	■ Filename - The name of the VPN batch file. ■ Realm - The Juniper VPN authentication realm. ■ URL - URL or IP address of the Juniper VPN gateway. ■ MIN - (true|false) Whether to run the BAT file minimized.
Outlook.bat	Configure Outlook's security profile to use the certificate for signing and encrypting. Supported on Windows only	No configuration is necessary to use this template.
WiFi.bat	Configure a certificate to connect to a Wi-Fi network. Supported on Windows only	■ ProfileName - The name of the WiFi profile (usually the same as the SSID, but on Windows XP or later, these can be different). This value is displayed to the user. ■ SSID - The SSID of your wireless network. ■ nonBroadcast - (true|false) Whether the network broadcasts the SSID. If set to true, the SSID is not broadcast. ■ connectionMode - (manual|auto) Whether the computer should automatically connect to this network when it is in range.

Appendix A

Certificate Properties Utility

This appendix includes the following topics:

About the Certificate Properties Utility

You can invoke the Certificate Properties utility from a script file to obtain the following information about a specific certificate:

- Expiration date and expiration status (expired or not)
- Subject and common name
- Issuer and serial number
- Policy object identifier (OID) for the certificate

Invoking the Certificate Properties Utility Usage (Windows)

To invoke the Certificate Properties utility (Windows)

1 Set the PKI directory in your BAT file using the CD command.

The second argument, %2, should always be the PKI directory in your BAT file. For example:

```
cd %2
```

This must be done only once in your BAT file unless some other operation uses the cd command. In that case, you must use the cd command to reset the directory.

2 Add the following commands to your BAT file:

```
tblive-4-helper-console-x86.exe DSM\cmdlineutil.dsm [operation]
[argument] [path to certificate file]
```

Table A-1 lists the operations and arguments that the command supports.

Table A-1 Certificate Properties Utility Options

Operation	Argument	Description
USAGE	None	Displays the utility usage and help file.
CERTPROPERTY	EXPIRED	Returns the expiration status of the certificate: ■ TRUE indicates that the certificate is expired. ■ FALSE indicates that the certificate is not expired.
	EXPIRATION_DATE	Returns the certificate expiration date. The date format is in the default system date format. To change the format of the response, add an argument to the end of the command to identify the date format. Use a date format string according to the C function strftime.

Table A-1 Certificate Properties Utility Options *(continued)*

Operation	Argument	Description
	SUBJECT	Returns the subject name of the certificate.
	SUBJECT_COMMON_NAME	Returns the common name in the certificate's subject name.
	ISSUER	Returns the name of the certificate issuer.
	SERIAL_NUMBER	Returns the certificate serial number.
	POLICY_OID	Returns the certificate's policy object identifier (OID). The policy OID should match the OID listed for the associated certificate profile in PKI Manager.

About the Certificate Properties Commands for Batch

The following examples describe how to use Certificate Properties commands. A full, contextual usage example is provided in the Sample Certificate Properties Utility Usage

See "Sample Certificate Properties Utility Usage" on page 16.

Note: Due to page limitations, the commands that are shown wrap to multiple lines. However, each command represents a single line. Separate commands in your BAT file should always be represented on one line.

- Obtaining the expired status of a certificate:

```
FOR /F "tokens=*" %%A IN ('tblive-4-helper-console-x86.exe
DSM\cmdlineutil.dsm CERTPROPERTY EXPIRED %3') DO SET Expired=%%A
```

- Obtaining the expiration date of a certificate:

```
FOR /F "tokens=*" %%A IN ('tblive-4-helper-console-x86.exe
DSM\cmdlineutil.dsm CERTPROPERTY EXPIRATION_DATE %3 "%%Y-%%m-%%d
%%H:%%M:%%S"') DO SET ExpirationDate=%%A
```

Note: You must use double percent characters when you run strftime commands in a BAT file. Also, make sure that you avoid using the ^ and & characters in date format strings.

- Obtaining the subject name of a certificate:

```
FOR /F "tokens=*" %%A IN ('tblive-4-helper-console-x86.exe
DSM\cmdlineutil.dsm CERTPROPERTY SUBJECT %3') DO SET Subject=%%A
```

- Obtaining the common name of a certificate:

```
FOR /F "tokens=*" %%A IN ('tblive-4-helper-console-x86.exe
DSM\cmdlineutil.dsm CERTPROPERTY SUBJECT_COMMON_NAME %3') DO SET
CertCommonName=%%A
```

- Obtaining the issuer of a certificate:

```
FOR /F "tokens=*" %%A IN ('tblive-4-helper-console-x86.exe
DSM\cmdlineutil.dsm
CERTPROPERTY ISSUER %3') DO SET Issuer=%%A
```

- Obtaining the serial number of a new certificate:

```
FOR /F "tokens=*" %%A IN ('tblive-4-helper-console-x86.exe
DSM\cmdlineutil.dsm CERTPROPERTY SERIAL_NUMBER %3') DO SET
SerialNumber=%%A
```

- Obtaining the policy OID of a certificate:

```
FOR /F "tokens=*" %%A IN ('tblive-4-helper-console-x86.exe
DSM\cmdlineutil.dsm CERTPROPERTY POLICY_OID %3') DO SET PolicyOID=%%A
```

Sample Certificate Properties Utility Usage

```
REM Initialize variables
SET Expired=
SET ERRORLEVEL=0

REM Change to PKI Client installation directory
CD %2

REM Call utility to determine expiration status
```

```
FOR /F "tokens=*" %%A IN (‘tblive-4-helper-console-x86.exe ^
DSM\cmdlineutil.dsm CERTPROPERTY EXPIRED %3’) DO SET ^ Expired=%%A

REM Check for errors from the utility
IF %ERRORLEVEL% GTR 0 EXIT /B %ERRORLEVEL%

REM Check for an empty result from the utility
IF "%Expired%"==”" EXIT /B 1000

REM Check the result, and quit if certificate is expired
IF "%Expired%"=="TRUE" EXIT /B 0

REM Ready for further processing.
ECHO Your certificate is valid.
```

Certificate Properties Utility Usage (Mac)

If you want to invoke the Certificate Properties utility, you must add the following commands to your SH file:

```
"$2"/tblive-4-helper-x86_64 -c cmdlineutil.dsm [operation] [argument]
[path to certificate file]
```

Table A-2 lists the operations and arguments that this command supports.

Table A-2 Certificate Properties Utility Options (Mac)

Operation	Argument	Description
USAGE	None	Displays the utility usage and help file.
CERTPROPERTY	EXPIRED	Returns the expiration status of the certificate: ■ TRUE indicates that the certificate is expired. ■ FALSE indicates that the certificate is not expired.

Table A-2 Certificate Properties Utility Options (Mac) *(continued)*

Operation	Argument	Description
	EXPIRATION_DATE	Returns the certificate expiration date. The date format is in the default system date format. To change the format of the response, add an argument to the end of the command to identify the date format. Use a date format string according to the C function strftime.
	SUBJECT	Returns the subject name of the certificate.
	SUBJECT_COMMON_NAME	Returns the common name in the certificate's subject name.
	ISSUER	Returns the name of the certificate issuer.
	SERIAL_NUMBER	Returns the certificate serial number.
	POLICY_OID	Returns the certificate's policy object identifier (OID). The policy OID should match the OID listed for the associated certificate profile in PKI Manager.

Certificate Properties Commands for Shell

The following examples describe how to apply the Certificate Properties commands. You can also review a full, contextual usage example.

See "Sample Certificate Properties Utility Usage" on page 19.

Due to page limitations, the commands that are shown wrap to multiple lines. However, each command represents a single line. Separate command in your SH file should always be represented on one line.

- Obtaining the expired status of a certificate:

```
Expired="$("$2"/tblive-4-helper-x86_64 -c cmdlineutil.dsm
CERTPROPERTY EXPIRED "$3")"
```

- Obtaining the expiration date of a certificate:

```
ExpirationDate="$("$2"/tblive-4-helper-x86_64 -c cmdlineutil.dsm
CERTPROPERTY EXPIRATION_DATE "$3" "%Y-%m-%d %H:%M:%S" )"
```

Note: You must use double percent characters when running strftime commands in an SH file. Also, avoid using the ^ and & characters in date format strings.

- Obtaining the subject name of a certificate:

```
Subject="$("$2"/tblive-4-helper-x86_64 -c cmdlineutil.dsm
CERTPROPERTY SUBJECT "$3")"
```

- Obtaining the common name of a certificate:

```
CertCommonName="$("$2"/tblive-4-helper-x86_64 -c cmdlineutil.dsm
CERTPROPERTY SUBJECT_COMMON_NAME "$3")"
```

- Obtaining the issuer of a certificate:

```
Issuer="$("$2"/tblive-4-helper-x86_64 -c cmdlineutil.dsm
CERTPROPERTY ISSUER "$3")"
```

- Obtaining the serial number of a new certificate:

```
SerialNumber="$("$2"/tblive-4-helper-x86_64 -c cmdlineutil.dsm
CERTPROPERTY SERIAL_NUMBER "$3")"
```

- Obtaining the policy OID of a certificate:

```
PolicyOID="$("$2"/tblive-4-helper-x86_64 -c cmdlineutil.dsm
CERTPROPERTY POLICY_OID "$3")"
```

Sample Certificate Properties Utility Usage

```
#!/bin/sh

# Call utility to determine expiration status
expired="$("$2"/tblive-4-helper-x86_64 -c cmdlineutil.dsm CERTPROPERTY
EXPIRED "$3")"

ErrorCheck=$?

# Check for errors from the utility
```

```
if [ $ErrorCheck -ne 0 ] ; then
    exit $ErrorCheck

# Check for an empty result from the utility
elif [ "$expired" == "" ] ; then
    exit 1

# Check the result, and quit if certificate is expired.
elif [ "$expired" == "TRUE" ] ; then
    exit 0
fi

# Ready for further processing
echo Your certificate is valid.
```

Appendix B

Error Codes and Troubleshooting

This appendix includes the following topics:

Script File Error Codes

The following error messages may occur when you process script files.

Table B-1 lists all the generic script file error messages.

Table B-1 Generic Script File Error Messages

Error Code	Description
2	Failed to get AES 256 CBC cipher.
3	Failed to initialize AES context.
4	Failed to initialize SHA256 hmac.
5	HMAC hash does not match.
6	Could not open script file. Solution: Verify that the script file is present in the correct location.
7	Script file does not match signature. Solution: The script file has been modified since it was signed.

Table B-1 Generic Script File Error Messages *(continued)*

Error Code	Description
8	Script file certificate is invalid. Solution: The certificate that is provided in the script file is either corrupt or does not conform to the policy for the available signing certificate chain.
9	Script file signing organization not trusted. Solution: The proper organization was not set in the policy. The corresponding log message contains the expected organization name.
10	Failed to get SHA256 digest.
11	Could not validate script certificate.

Table B-2 lists all certificate properties utility error messages.

Table B-2 Certificate Properties Utility Error Messages

Error Code	Description
50	Command line utility cannot open certificate file.
51	Invalid certificate property that is provided to command line utility.
52	Command line utility cannot find the specified value.
53	Command line utility cannot parse certificate file.
54	Invalid operation for command line utility.

The following error messages are specific to individual script files. Symantec recommends that you use unique error codes when you create custom scripts. For example, you can use four-digit error codes that start at 1000.

Table B-3 lists all InstallCA Signed script file error messages.

Table B-3 InstallCA.Signed.Script File Error Messages

Error Code	Description
100	Could not open certificate file.
101	Could not open Current User's Root certificate store.
102	Could not create certificate context for root certificate.

Table B-3 InstallCA.Signed.Script File Error Messages *(continued)*

Error Code	Description
103	Could not add root certificate to store.
104	Could not open Current User's Intermediate certificate store.
105	Failed to parse intermediate certificate.
106	Could not create certificate context for intermediate certificate.
107	Could not add certificate to store.

Table B-4 lists all RegisterFirefox Signed script file error messages.

Table B-4 RegisterFirefox.Signed.Script File Error Messages

Error Code	Description
193	Could not rebuild Mozilla Firefox PKCS11 database
194	NSS_DEFAULT_DB_TYPE environment variable not set
196	Could not verify installation of PKCS11 module.
197	Could not locate path to Firefox profiles.
198	Could not read Firefox Install Directory from registry.
199	Could not read Firefox CurrentVersion from registry.
200	Could not add the Certificate Authority Root to Firefox.
201	An error occurred in Firefox's NSS utilities.

Table B-5 lists ADPub Signed script file error messages.

Table B-5 ADPub.Signed.Script File Error Messages

Error Code	Description
300	Failed to parse certificate data.
301	Failure in LDAP Update.

Table B-6 lists Microsoft Outlook Signed script file error messages.

Table B-6 Outlook.Signed.Script File Error Messages

Error Code	Description
400	Could not open Microsoft Office from registry.
401	Could not open Outlook from registry.
402	Failure in Outlook MAPI.
403	Failed to launch 64-bit process.
404	Unable to retrieve a valid bitness value from registry.
405	Failed to parse certificate data.

Table B-7 lists Juniper Signed script file error messages.

Table B-7 Juniper.Signed.Script File Error Messages

Error Code	Description
500	Could not parse certificate
501	Could not find certificate common name.
502	Could not open Network Connect from registry
503	Could not read InstallPath from registry.
504	Could not retrieve Policy OID from certificate
505	Missing authentication realm in configuration file
506	Missing URL in configuration file
507	Missing file name in configuration file

Table B-8 lists WiFi Signed script file error messages.

Table B-8 WiFi.Signed.Script File Error Messages

Error Code	Description
600	Could not find Profile Name in configuration file
601	Could not find network SSID in configuration file
602	Could not find valid nonBroadcast in configuration file (true/false)
603	Could not find valid connectionMode in configuration file (auto/manual)

Table B-8 WiFi.Signed.Script File Error Messages *(continued)*

Error Code	Description
604	Invalid wireless profile XML
605	Wireless profile cannot be saved

Table B-9 lists Cisco Signed script file error messages.

Table B-9 Cisco.Signed.Script File Error Messages

Error Code	Description
700	Could not retrieve certificate policy OID
701	Could not find certificate common name.
702	Could not find certificate subject
703	Could not find Profile Name in configuration file
704	Could not find Host in configuration file
705	Failed to create Cisco VPN profile

Index

M

O

P

R

S

T

U

W