NAME

MakeCert – Create X.509 certificates for test purposes

SYNOPSIS

makecert [options] certificate

DESCRIPTION

Create an X.509 certificate using the provided informations. This is useful for testing Authenticode signatures, SSL and S/MIME technologies.

PARAMETERS

-# num Specify the certificate serial number.

-n dn Specify the subject Distinguished Name (DN).

-in dn Specify the issuer Distinguished Name (DN).

-r Create a self-signed, also called root, certificate.

-iv pvkfile
Specify the private key file (.PVK) for the issuer. The private key in the specified file will be used to sign the new certificate.

-ic certfile
Extract the issuer's name from the specified certificate file - i.e. the subject name of the specified certificate becomes the issuer name of the new certificate.

-in name
Use the issuer's name from the specified parameter.

-ik container
Specify the key container name to be used for the issuer.

-iky [signature | exchange | #]
Specify the key number to be used in the provider (when used with -ik).

-ip provider
Specify the cryptographic provider to be used for the issuer.

-ir [localmachine | currentuser]
Specify the provider will search the user or the machine keys containers for the issuer.

-iy number
Specify the provider type to be used for the issuer.

-sv pkvfile
Specify the private key file (.PVK) for the subject. The public part of the key will be inserted into the created certificate. If non-existant the specified file will be created with a new key pair (default to 1024 bits RSA key pair).

-sk container
Specify the key container name to be used for the subject.

-sky [signature | exchange | #]
Specify the key number to be used in the provider (when used with -sk).

-sp provider
Specify the cryptographic provider to be used for the subject.

-sr [localmachine | currentuser]
Specify the provider will search the user or the machine keys containers for the subject.

-sy number
Specify the provider type to be used for the issuer.

-a hash Select hash algorithm. Only MD5 and SHA1 algorithms are supported.

-b date The date since when the certificate is valid (notBefore).

-e date The date until when the certificate is valid (notAfter).

-m number
Specify the certificate validity period in months. This is added to the notBefore validity date which can be set with -b or will default to the current date/time.

-cy [authority|end]
Basic constraints. Select Authority or End-Entity certificate. Only Authority certificates can be used to sign other certificates (-ic). End-Entity can be used by clients (e.g. Authenticode, S/MIME) or servers (e.g. SSL).

-h number
Add a path length restriction to the certificate chain. This is only applicable for certificates that have BasicConstraint set to Authority (-cy authority). This is used to limit the chain of certificates than can be issued under this authority.

-alt filename
Add a subjectAltName extension to the certificate. Each line from ’filename’ will be added as a DNS entry of the extension. This option is useful if you want to create a single SSL certificate to work on several hosts that do not share a common domain name (i.e. CN=*.domain.com would not work).

-eku oid[,oid]
Add some extended key usage OID to the certificate.

-p12 pkcs12file password
Create a new PKCS#12 file containing both the certificates (the subject and possibly the issuer’s) and the private key. The PKCS#12 file is protected with the specified password. This option is **mono exclusive.**

-? Help (display this help message)

-! Extended help (for advanced options)

EXAMPLES

To create a SSL test (i.e. non trusted) certificate is easy once your know your host’s name. The following command will create a test certificate for an SSL server:

```
$ hostname
pollux

$ makecert -r -eku 1.3.6.1.5.5.7.3.1 -n "CN=pollux" -sv pollux.pvk pollux.cer
Success
```

In particular in the above example, the parameters used to build this test certificate were:

-r Create a self-signed certificate (i.e. without an hierarchy).

-eku 1.3.6.1.5.5.7.3.1
Optional (as sadly most client don’t require it). This indicates that your certificate is intended for server-side authentication.

-n Common Name (CN) = Host name. This is verified the SSL client and must match the connected host (or else you’ll get a warning or error or *gasp* nothing).

-sv private.key
The private key file. The key (1024 bits RSA key pair) will be automatically generated if the specified file isn’t present.

pollux.cer
The SSL certificate to be created for your host.

KNOWN RESTRICTIONS

Compared to the Windows version some options aren't supported (-$, -d, -l, -nscp, -is, -sc, -ss). Also PVK files with passwords aren't supported.

AUTHOR

Written by Sebastien Pouliot

COPYRIGHT

MAILING LISTS

Visit http://lists.ximian.com/mailman/listinfo/mono-devel-list for details.

WEB SITE

Visit http://www.mono-project.com for details

SEE ALSO

signcode(1)