mk-ca-bundle the man page

NAME

mk-ca-bundle - convert mozilla's certdata.txt to PEM format

SYNOPSIS

mk-ca-bundle [options] *[outputfile]*

DESCRIPTION

The mk-ca-bundle tool downloads the certdata.txt file from Mozilla's source tree over HTTPS, then parses certdata.txt and extracts certificates into PEM format. By default, only CA root certificates trusted to issue SSL server authentication certificates are extracted. These are then processed with the OpenSSL commandline tool to produce the final ca-bundle file.

The default *outputfile* name is **ca-bundle.crt**. By setting it to '-' (a single dash) you will get the output sent to STDOUT instead of a file.

The PEM format this scripts uses for output makes the result readily available for use by just about all OpenSSL or GnuTLS powered applications, such as curl, wget and more.

OPTIONS

The following options are supported:

-b

backup an existing version of *outputfilename*

-d [name]

specify which Mozilla tree to pull certdata.txt from (or a custom URL). Valid names are: aurora, beta, central, mozilla, nss, release (default). They are shortcuts for which source tree to get the cert data from.

-f

force rebuild even if certdata.txt is current (Added in version 1.17)

-i

print version info about used modules

-k

Allow insecure data transfer. By default (since 1.27) this command will fail if the HTTPS transfer fails. This overrides that decision (and opens for man-in-the-middle attacks).

-l

print license info about certdata.txt

-m

(Added in 1.26) Include meta data comments in the output. The meta data is specific information about each certificate that is stored in the original file as comments and using this option will make those comments get passed on to the output file. The meta data is not parsed in any way by mk-ca-bundle.

-n

no download of certdata.txt (to use existing)

-p [purposes]:[levels]

list of Mozilla trust purposes and levels for certificates to include in output. Takes the form of a comma separated list of purposes, a colon, and a comma separated list of levels. The default is to include all certificates trusted to issue SSL Server certificates (SERVER_AUTH:TRUSTED_DELEGATOR).

(Added in version 1.21, Perl only)

Valid purposes are:

> ALL, DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCIPHERMENT, DATA_ENCIPHERMENT, KEY_AGREEMENT, KEY_CERT_SIGN, CRL_SIGN, SERVER_AUTH (default), CLIENT_AUTH, CODE_SIGNING, EMAIL_PROTECTION, IPSEC_END_SYSTEM, IPSEC_TUNNEL, IPSEC_USER, TIME_STAMPING, STEP_UP_APPROVED

Valid trust levels are:

> ALL, TRUSTED_DELEGATOR (default), NOT_TRUSTED, MUST_VERIFY_TRUST, TRUSTED

-q

be really quiet (no progress output at all)

-t

include plain text listing of certificates

-s [algorithms]

comma separated list of signature algorithms with which to hash/fingerprint each certificate and output when run in plain text mode.

(Added in version 1.21, Perl only)

Valid algorithms are:

ALL, NONE, MD5 (default), SHA1, SHA256, SHA384, SHA512

-u

unlink (remove) certdata.txt after processing

-v

be verbose and print out processed CAs

EXIT STATUS

Returns 0 on success. Returns 1 if it fails to download data.

CERTDATA FORMAT

The file format used by Mozilla for this trust information seems to be documented here:

`https://p11-glue.freedesktop.org/doc/storing-trust-policy/stor`