```
#
===========================================================================
===
# Licensed Materials - Property of IBM
# "Restricted Materials of IBM"
#
# IBM SDK, Java(tm) Technology Edition, v8
# (C) Copyright IBM Corp. 2010, 2022. All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or disclosure
# restricted by GSA ADP Schedule Contract with IBM Corp.
#
===========================================================================
===
#
# This is the "master security properties file".
#
# An alternate java.security properties file may be specified
# from the command line via the system property
#
#    -Djava.security.properties=<URL>
#
# This properties file appends to the master security properties file.
# If both properties files specify values for the same key, the value
# from the command-line properties file is selected, as it is the last
# one loaded.
#
# Also, if you specify
#
#    -Djava.security.properties==<URL> (2 equals),
#
# then that properties file completely overrides the master security
# properties file.
#
# To disable the ability to specify an additional properties file from
# the command line, set the key security.overridePropertiesFile
# to false in the master security properties file. It is set to true
# by default.

# In this file, various security properties are set for use by
# java.security classes. This is where users can statically register
# Cryptography Package Providers ("providers" for short). The term
# "provider" refers to a package or set of packages that supply a
# concrete implementation of a subset of the cryptography aspects of
# the Java Security API. A provider may, for example, implement one or
# more digital signature algorithms or message digest algorithms.
#
# Each provider must implement a subclass of the Provider class.
# To register a provider in this master security properties file,
# specify the Provider subclass name and priority in the format
#
#    security.provider.<n>=<className>
#
# This declares a provider, and specifies its preference
# order n. The preference order is the order in which providers are
# searched for requested algorithms (when no specific provider is
# requested). The order is 1-based; 1 is the most preferred, followed
# by 2, and so on.
#
# <className> must specify the subclass of the Provider class whose
# constructor sets the values of various properties that are required
# for the Java Security API to look up the algorithms or other
# facilities implemented by the provider.
#
# There must be at least one provider specification in java.security.
# There is a default provider that comes standard with the JDK. It
# is called the "SUN" provider, and its Provider subclass
# named Sun appears in the sun.security.provider package. Thus, the
# "SUN" provider is registered via the following:
#
#    security.provider.1=sun.security.provider.Sun
#
# (The number 1 is used for the default provider.)
#
# Note: Providers can be dynamically registered instead by calls to
# either the addProvider or insertProviderAt method in the Security
# class.

#
# List of providers and their preference orders (see above):
#
security.provider.1=com.ibm.jsse2.IBMJSSEProvider2
security.provider.2=com.ibm.crypto.plus.provider.IBMJCEPlus
security.provider.3=com.ibm.crypto.provider.IBMJCE
security.provider.4=com.ibm.security.jgss.IBMJGSSProvider
security.provider.5=com.ibm.security.cert.IBMCertPath
security.provider.6=com.ibm.security.sasl.IBMSASL
security.provider.7=com.ibm.xml.crypto.IBMXMLCryptoProvider
security.provider.8=com.ibm.xml.enc.IBMXMLEncProvider
security.provider.9=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
security.provider.10=sun.security.provider.Sun
security.provider.11=com.ibm.i5os.jsse.JSSEProvider

#
# IBMJCE and IBMSecureRandom SecureRandom seed source.
#
# Select the primary source of seed data for the "SHA1PRNG" and
# "NativePRNG" SecureRandom implementations in the "IBMJCE"
# provider and the "SHA1PRNG" SecureRandom implementation
# in the "IBMSecureRandom" provider.
# (Other SecureRandom implementations might also use this property.)
#
# On Unix-like systems (for example, Solaris/Linux/MacOS), the
# "NativePRNG" and "SHA1PRNG" implementations obtains seed data from
# special device files such as file:/dev/random.
#
# On Windows systems, specifying the URLs "file:/dev/random" or
# "file:/dev/urandom" will enable the native Microsoft CryptoAPI seeding
# mechanism for SHA1PRNG.
#
# By default, an attempt is made to use the entropy gathering device
# specified by the "securerandom.source" Security property.  If an
# exception occurs while accessing the specified URL:
#
#     SHA1PRNG:
#         the traditional system/thread activity algorithm will be used.
#
#     NativePRNG:
#         a default value of /dev/random will be used.  If neither
#         are available, the implementation will be disabled.
#         "file" is the only currently supported protocol type.
#
# The entropy gathering device can also be specified with the System
# property "java.security.egd". For example:
#
#   % java -Djava.security.egd=file:/dev/random MainClass
#
# Specifying this System property will override the
# "securerandom.source" Security property.
#
# In addition, if "file:/dev/random" or "file:/dev/urandom" is
# specified, the "NativePRNG" implementation will be more preferred than
# SHA1PRNG in the IBMJCE provider.
#
securerandom.source=file:/dev/urandom

#
# A list of known strong SecureRandom implementations.
#
# To help guide applications in selecting a suitable strong
# java.security.SecureRandom implementation, Java distributions should
# indicate a list of known strong implementations using the property.
#
# This is a comma-separated list of algorithm and/or algorithm:provider
# entries.
#
securerandom.strongAlgorithms=SHA2DRBG:IBMJCE

#
# Class to instantiate as the javax.security.auth.login.Configuration
# provider.
#
login.configuration.provider=com.ibm.security.auth.login.ConfigFile

#
# Default login configuration file
#
#login.config.url.1=file:${user.home}/.java.login.config

#
# Class to instantiate as the system Policy. This is the name of the class
# that will be used as the Policy object.
#
policy.provider=sun.security.provider.PolicyFile

# The default is to have a single system-wide policy file,
# and a policy file in the user's home directory.
policy.url.1=file:${java.home}/lib/security/java.policy
policy.url.2=file:${java.home}/lib/security/java.pol
policy.url.3=file:///${user.home}/.java.policy

# whether or not we expand properties in the policy file
# if this is set to false, properties (${...}) will not be expanded in policy
# files.
policy.expandProperties=true
```

```
# whether or not we allow an extra policy to be passed on the command line
# with -Djava.security.policy=somefile. Comment out this line to disable
# this feature.
policy.allowSystemProperty=true

# whether or not we look into the IdentityScope for trusted Identities
# when encountering a 1.1 signed JAR file. If the identity is found
# and is trusted, we grant it AllPermission.
policy.ignoreIdentityScope=false

#
# Default keystore type.
#
keystore.type=jks

#
# Controls compatibility mode for the JKS keystore type.
#
# When set to 'true', the JKS keystore type supports loading
# keystore files in either JKS or PKCS12 format. When set to 'false'
# it supports loading only JKS keystore files.
#
keystore.type.compat=true

#
# List of comma-separated packages that start with or equal this string
# will cause a security exception to be thrown when
# passed to checkPackageAccess unless the
# corresponding RuntimePermission ("accessClassInPackage."+package) has
# been granted.
package.access=sun.,\
               com.ibm.oti.,\
               openj9.internal.,\
               com.intel.fortress.,\
               com.sun.xml.internal.,\
               com.sun.imageio.,\
               com.sun.istack.internal.,\
               com.sun.jmx.,\
               com.sun.media.sound.,\
               com.sun.naming.internal.,\
               com.sun.proxy.,\
               com.sun.corba.se.,\
               com.sun.org.apache.bcel.internal.,\
               com.sun.org.apache.regexp.internal.,\
               com.sun.org.apache.xerces.internal.,\
               com.sun.org.apache.xpath.internal.,\
               com.sun.org.apache.xalan.internal.extensions.,\
               com.sun.org.apache.xalan.internal.lib.,\
               com.sun.org.apache.xalan.internal.res.,\
               com.sun.org.apache.xalan.internal.templates.,\
               com.sun.org.apache.xalan.internal.utils.,\
               com.sun.org.apache.xalan.internal.xslt.,\
               com.sun.org.apache.xalan.internal.xsltc.cmdline.,\
               com.sun.org.apache.xalan.internal.xsltc.compiler.,\
               com.sun.org.apache.xalan.internal.xsltc.trax.,\
               com.sun.org.apache.xalan.internal.xsltc.util.,\
               com.sun.org.apache.xml.internal.res.,\
               com.sun.org.apache.xml.internal.security.,\
               com.sun.org.apache.xml.internal.serializer.utils.,\
               com.sun.org.apache.xml.internal.utils.,\
               com.sun.org.glassfish.,\
               com.oracle.xmlns.internal.,\
               com.oracle.webservices.internal.,\
               com.ibm.stax.,\
               com.ibm.xml.jaxp.datatype.,\
               com.ibm.xml.resolver.,\
               com.ibm.xml.xlxp.api.event.,\
               com.ibm.xml.xlxp.api.jaxp.impl.,\
               com.ibm.xml.xlxp.api.sax.impl.,\
               com.ibm.xml.xlxp.api.stax.events.,\
               com.ibm.xml.xlxp.api.stax.msg.,\
               com.ibm.xml.xlxp.api.stax.serializer.,\
               com.ibm.xml.xlxp.api.util.,\
               com.ibm.xml.xlxp.scan.msg.,\
               com.ibm.xml.xlxp.scan.util.,\
               com.ibm.xtq.ast.parsers.xpath.,\
               com.ibm.xtq.ast.parsers.xslt.,\
               com.ibm.xtq.ast.res.,\
               com.ibm.xtq.ast.visit.,\
               com.ibm.xtq.bcel.,\
               com.ibm.xtq.common.utils.,\
               com.ibm.xtq.utils.,\
               com.ibm.xtq.xml.datamodel.,\
               com.ibm.xtq.xml.dtm.ref.sax2dtm.,\
               com.ibm.xtq.xml.dtm.utils.,\
               com.ibm.xtq.xml.experimental.,\
               com.ibm.xtq.xml.res.,\
               com.ibm.xtq.xml.types.,\
               com.ibm.xtq.xml.unicode.normalize.,\
               com.ibm.xtq.xml.utils.,\
               com.ibm.xtq.xml.xdm.dom.,\
               com.ibm.xtq.xml.xdm.ref.,\
               com.ibm.xtq.xml.xdm.res.,\
               com.ibm.xtq.xpath.jaxp.,\
               com.ibm.xtq.xslt.cmdline.,\
               com.ibm.xtq.xslt.jaxp.interpreter.,\
               com.ibm.xtq.xslt.res.,\
               com.ibm.xtq.xslt.runtime.debug.,\
               com.ibm.xtq.xslt.runtime.output.,\
               com.ibm.xtq.xslt.runtime.res.,\
               com.ibm.xtq.xslt.runtime.v2.,\
               com.ibm.xtq.xslt.translator.v1.,\
               com.ibm.xtq.xslt.translator.v2.,\
               com.ibm.xtq.xslt.typechecker.,\
               com.ibm.xtq.xslt.xylem.autof.,\
               com.ibm.xtq.xslt.xylem.codegen.,\
               com.ibm.xtq.xslt.xylem.interpreter.,\
               com.ibm.xtq.xslt.xylem.optimizers.,\
               com.ibm.xtq.xslt.xylem.parser.,\
               com.ibm.xtq.xslt.xylem.partialeval.,\
               com.ibm.xtq.xslt.xylem.types.,\
               com.ibm.xtq.xslt.xylem.xpath20.analysis.,\
               com.ibm.xtq.xslt.xylem.xpath20.parser.,\
               com.ibm.xtq.xslt.xylem.xpath20.typesystem.,\
               com.ibm.xylem.annot.meta.,\
               com.ibm.xylem.builders.,\
               com.ibm.xylem.codegen.,\
               com.ibm.xylem.commandline.,\
               com.ibm.xylem.config.,\
               com.ibm.xylem.drivers.,\
               com.ibm.xylem.interpreter.,\
               com.ibm.xylem.parser.,\
               com.ibm.xylem.res.,\
               com.ibm.xylem.types.,\
               com.ibm.xylem.utils.,\
               com.sun.org.apache.xalan.internal.xsltc.trax.,\
               com.sun.org.apache.xerces.internal.dom.,\
               com.sun.org.apache.xerces.internal.jaxp.,\
               com.sun.org.apache.xerces.internal.parsers.,\
               com.sun.org.apache.xpath.internal.jaxp.,\
               com.sun.xml.internal.stream.,\
               org.apache.html.dom.,\
               org.apache.wml.,\
               org.apache.xalan.client.,\
               org.apache.xalan.extensions.,\
               org.apache.xalan.lib.sql.,\
               org.apache.xalan.res.,\
               org.apache.xalan.serialize.,\
               org.apache.xalan.templates.,\
               org.apache.xalan.trace.,\
               org.apache.xalan.transformer.,\
               org.apache.xalan.xslt.,\
               org.apache.xalan.xsltc.cmdline.,\
               org.apache.xerces.dom.events.,\
               org.apache.xerces.dom3.as.,\
               org.apache.xerces.impl.dtd.,\
               org.apache.xerces.impl.dv.util.,\
               org.apache.xerces.impl.io.,\
               org.apache.xerces.impl.msg.,\
               org.apache.xerces.impl.validation.,\
               org.apache.xerces.impl.xpath.,\
               org.apache.xerces.impl.xs.,\
               org.apache.xerces.util.,\
               org.apache.xerces.xinclude.,\
               org.apache.xerces.xni.grammars.,\
               org.apache.xerces.xpointer.,\
               org.apache.xerces.xs.datatypes.,\
               org.apache.xml.dtm.ref.dom2dtm.,\
               org.apache.xml.dtm.ref.sax2dtm.,\
               org.apache.xml.res.,\
               org.apache.xml.serializer.charmap.,\
               org.apache.xml.serializer.dom3.,\
               org.apache.xml.serializer.unicode.,\
               org.apache.xml.serializer.utils.,\
               org.apache.xml.utils.,\
               org.apache.xmlcommons.,\
               org.apache.xpath.axes.,\
               org.apache.xpath.compiler.,\
               org.apache.xpath.functions.,\
               org.apache.xpath.objects.,\
               org.apache.xpath.operations.,\
               org.apache.xpath.patterns.,\
               org.apache.xpath.res.,\
               oracle.jrockit.jfr.,\
               org.jcp.xml.dsig.internal.,\
               com.ibm.rmi.channel.,\
                              com.ibm.rmi.corba.,\
                              com.ibm.rmi.iiop.,\
                              com.ibm.rmi.io.,\
                              com.ibm.rmi.pi.,\
                              com.ibm.rmi.poa.,\
                              com.ibm.rmi.ras.,\
                              com.ibm.rmi.transport.,\
                              com.ibm.rmi.util.,\
                              com.ibm.CORBA.channel.orb.,\
                              com.ibm.CORBA.iiop.,\
                              com.ibm.CORBA.nio.,\
                              com.ibm.CORBA.poa.,\
                              com.ibm.CORBA.transport.,\
                              jdk.internal.,\
                              jdk.nashorn.internal.,\
                              jdk.nashorn.tools.,\
                              jdk.xml.internal.,\
                              com.sun.activation.registries.,\
                              com.sun.browser.,\
                              com.sun.glass.,\
                              com.sun.javafx.,\
                              com.sun.media.,\
                              com.sun.openpisces.,\
                              com.sun.prism.,\
                              com.sun.scenario.,\
                              com.sun.t2k.,\
                              com.sun.pisces.,\
                              com.sun.webkit.,\
                              jdk.management.resource.internal.

               #
               # List of comma-separated packages that start with or equal this string
               # will cause a security exception to be thrown when
               # passed to checkPackageDefinition unless the
               # corresponding RuntimePermission ("defineClassInPackage."+package) has
               # been granted.
               #
               # by default, none of the class loaders supplied with the JDK call
               # checkPackageDefinition.
               #
               package.definition=sun.,\
                                  com.sun.xml.internal.,\
                                  com.sun.imageio.,\
                                  com.sun.istack.internal.,\
               com.sun.jmx.,\
               com.sun.media.sound.,\
               com.sun.naming.internal.,\
               com.sun.proxy.,\
               com.sun.corba.se.,\
               com.sun.org.apache.bcel.internal.,\
               com.sun.org.apache.regexp.internal.,\
               com.sun.org.apache.xerces.internal.,\
               com.sun.org.apache.xpath.internal.,\
               com.sun.org.apache.xalan.internal.extensions.,\
               com.sun.org.apache.xalan.internal.lib.,\
               com.sun.org.apache.xalan.internal.res.,\
               com.sun.org.apache.xalan.internal.templates.,\
               com.sun.org.apache.xalan.internal.utils.,\
               com.sun.org.apache.xalan.internal.xslt.,\
               com.sun.org.apache.xalan.internal.xsltc.cmdline.,\
               com.sun.org.apache.xalan.internal.xsltc.compiler.,\
               com.sun.org.apache.xalan.internal.xsltc.trax.,\
               com.sun.org.apache.xalan.internal.xsltc.util.,\
               com.sun.org.apache.xml.internal.res.,\
               com.sun.org.apache.xml.internal.security.,\
               com.sun.org.apache.xml.internal.serializer.utils.,\
               com.sun.org.apache.xml.internal.utils.,\
               com.sun.org.glassfish.,\
               com.oracle.xmlns.internal.,\
               com.oracle.webservices.internal.,\
               com.ibm.stax.,\
               com.ibm.xml.jaxp.datatype.,\
               com.ibm.xml.resolver.,\
               com.ibm.xml.xlxp.api.event.,\
               com.ibm.xml.xlxp.api.jaxp.impl.,\
               com.ibm.xml.xlxp.api.sax.impl.,\
               com.ibm.xml.xlxp.api.stax.events.,\
               com.ibm.xml.xlxp.api.stax.msg.,\
               com.ibm.xml.xlxp.api.stax.serializer.,\
               com.ibm.xml.xlxp.api.util.,\
               com.ibm.xml.xlxp.scan.msg.,\
               com.ibm.xml.xlxp.scan.util.,\
               com.ibm.xtq.ast.parsers.xpath.,\
               com.ibm.xtq.ast.parsers.xslt.,\
               com.ibm.xtq.ast.res.,\
               com.ibm.xtq.ast.visit.,\
               com.ibm.xtq.bcel.,\
               com.ibm.xtq.common.utils.,\
               com.ibm.xtq.utils.,\
               com.ibm.xtq.xml.datamodel.,\
               com.ibm.xtq.xml.dtm.ref.sax2dtm.,\
               com.ibm.xtq.xml.dtm.utils.,\
               com.ibm.xtq.xml.experimental.,\
               com.ibm.xtq.xml.res.,\
               com.ibm.xtq.xml.types.,\
               com.ibm.xtq.xml.unicode.normalize.,\
               com.ibm.xtq.xml.utils.,\
               com.ibm.xtq.xml.xdm.dom.,\
               com.ibm.xtq.xml.xdm.ref.,\
               com.ibm.xtq.xml.xdm.res.,\
               com.ibm.xtq.xpath.jaxp.,\
               com.ibm.xtq.xslt.cmdline.,\
               com.ibm.xtq.xslt.jaxp.interpreter.,\
               com.ibm.xtq.xslt.res.,\
               com.ibm.xtq.xslt.runtime.debug.,\
               com.ibm.xtq.xslt.runtime.output.,\
               com.ibm.xtq.xslt.runtime.res.,\
               com.ibm.xtq.xslt.runtime.v2.,\
               com.ibm.xtq.xslt.translator.v1.,\
               com.ibm.xtq.xslt.translator.v2.,\
               com.ibm.xtq.xslt.typechecker.,\
               com.ibm.xtq.xslt.xylem.autof.,\
               com.ibm.xtq.xslt.xylem.codegen.,\
               com.ibm.xtq.xslt.xylem.interpreter.,\
               com.ibm.xtq.xslt.xylem.optimizers.,\
               com.ibm.xtq.xslt.xylem.parser.,\
               com.ibm.xtq.xslt.xylem.partialeval.,\
               com.ibm.xtq.xslt.xylem.types.,\
               com.ibm.xtq.xslt.xylem.xpath20.analysis.,\
               com.ibm.xtq.xslt.xylem.xpath20.parser.,\
               com.ibm.xtq.xslt.xylem.xpath20.typesystem.,\
               com.ibm.xylem.annot.meta.,\
               com.ibm.xylem.builders.,\
               com.ibm.xylem.codegen.,\
               com.ibm.xylem.commandline.,\
               com.ibm.xylem.config.,\
               com.ibm.xylem.drivers.,\
               com.ibm.xylem.interpreter.,\
               com.ibm.xylem.parser.,\
               com.ibm.xylem.res.,\
               com.ibm.xylem.types.,\
               com.ibm.xylem.utils.,\
               com.sun.org.apache.xalan.internal.xsltc.trax.,\
               com.sun.org.apache.xerces.internal.dom.,\
               com.sun.org.apache.xerces.internal.jaxp.,\
               com.sun.org.apache.xerces.internal.parsers.,\
               com.sun.org.apache.xpath.internal.jaxp.,\
               com.sun.xml.internal.stream.,\
               org.apache.html.dom.,\
               org.apache.wml.,\
               org.apache.xalan.client.,\
               org.apache.xalan.extensions.,\
               org.apache.xalan.lib.sql.,\
               org.apache.xalan.res.,\
               org.apache.xalan.serialize.,\
               org.apache.xalan.templates.,\
               org.apache.xalan.trace.,\
               org.apache.xalan.transformer.,\
               org.apache.xalan.xslt.,\
               org.apache.xalan.xsltc.cmdline.,\
               org.apache.xerces.dom.events.,\
               org.apache.xerces.dom3.as.,\
               org.apache.xerces.impl.dtd.,\
               org.apache.xerces.impl.dv.util.,\
               org.apache.xerces.impl.io.,\
               org.apache.xerces.impl.msg.,\
               org.apache.xerces.impl.validation.,\
               org.apache.xerces.impl.xpath.,\
               org.apache.xerces.impl.xs.,\
               org.apache.xerces.util.,\
               org.apache.xerces.xinclude.,\
               org.apache.xerces.xni.grammars.,\
               org.apache.xerces.xpointer.,\
               org.apache.xerces.xs.datatypes.,\
               org.apache.xml.dtm.ref.dom2dtm.,\
               org.apache.xml.dtm.ref.sax2dtm.,\
               org.apache.xml.res.,\
               org.apache.xml.serializer.charmap.,\
               org.apache.xml.serializer.dom3.,\
               org.apache.xml.serializer.unicode.,\
               org.apache.xml.serializer.utils.,\
               org.apache.xml.utils.,\
               org.apache.xmlcommons.,\
               org.apache.xpath.axes.,\
               org.apache.xpath.compiler.,\
               org.apache.xpath.functions.,\
                org.apache.xpath.objects.,\
                org.apache.xpath.operations.,\
                org.apache.xpath.patterns.,\
                org.apache.xpath.res.,\
                oracle.jrockit.jfr.,\
                org.jcp.xml.dsig.internal.,\
                jdk.internal.,\
                jdk.nashorn.internal.,\
                jdk.nashorn.tools.,\
                jdk.xml.internal.,\
                com.sun.activation.registries.,\
                com.sun.browser.,\
                com.sun.glass.,\
                com.sun.javafx.,\
                com.sun.media.,\
                com.sun.openpisces.,\
                com.sun.prism.,\
                com.sun.scenario.,\
                com.sun.t2k.,\
                com.sun.pisces.,\
                com.sun.webkit.,\
                jdk.management.resource.internal.

#
# Determines whether this properties file can be appended to
# or overridden on the command line via -Djava.security.properties
#
security.overridePropertiesFile=true

#
# Determines the default key and trust manager factory algorithms for
# the javax.net.ssl package.
#
ssl.KeyManagerFactory.algorithm=IbmX509
ssl.TrustManagerFactory.algorithm=PKIX

#
# Other key and trust manager factory providers.
#
# IBM i5/OS key and trust manager factory
# ssl.KeyManagerFactory.algorithm=IbmISeriesX509
# ssl.TrustManagerFactory.algorithm=IbmISeriesX509

# IBM i5/OS SSL socket factory and SSL server socket factory providers
# ssl.SocketFactory.provider=com.ibm.i5os.jsse.JSSESocketFactory
# ssl.ServerSocketFactory.provider=com.ibm.i5os.jsse.JSSEServerSocketFactory

#
# The Java-level namelookup cache policy for successful lookups:
#
# any negative value: caching forever
# any positive value: the number of seconds to cache an address for
# zero: do not cache
#
# default value is forever (FOREVER). For security reasons, this
# caching is made forever when a security manager is set. When a security
# manager is not set, the default behavior in this implementation
# is to cache for 30 seconds.
#
# NOTE: setting this to anything other than the default value can have
#       serious security implications. Do not set it unless
#       you are sure you are not exposed to DNS spoofing attack.
#
#networkaddress.cache.ttl=-1

# The Java-level namelookup cache policy for failed lookups:
#
# any negative value: cache forever
# any positive value: the number of seconds to cache negative lookup results
# zero: do not cache
#
# In some Microsoft Windows networking environments that employ
# the WINS name service in addition to DNS, name service lookups
# that fail may take a noticeably long time to return (approx. 5 seconds).
# For this reason the default caching policy is to maintain these
# results for 10 seconds.
#
#
networkaddress.cache.negative.ttl=10

#
# Properties to configure OCSP for certificate revocation checking
#

# Enable OCSP
#
# By default, OCSP is not used for certificate revocation checking.
# This property enables the use of OCSP when set to the value "true".
```

```
#
# NOTE: SocketPermission is required to connect to an OCSP responder.
#
# Example,
#   ocsp.enable=true

#
# Location of the OCSP responder
#
# By default, the location of the OCSP responder is determined implicitly
# from the certificate being validated. This property explicitly specifies
# the location of the OCSP responder. The property is used when the
# Authority Information Access extension (defined in RFC 3280) is absent
# from the certificate or when it requires overriding.
#
# Example,
#   ocsp.responderURL=http://ocsp.example.net:80

#
# Subject name of the OCSP responder's certificate
#
# By default, the certificate of the OCSP responder is that of the issuer
# of the certificate being validated. This property identifies the certificate
# of the OCSP responder when the default does not apply. Its value is a string
# distinguished name (defined in RFC 2253) which identifies a certificate in
# the set of certificates supplied during cert path validation. In cases where
# the subject name alone is not sufficient to uniquely identify the certificate
# then both the "ocsp.responderCertIssuerName" and
# "ocsp.responderCertSerialNumber" properties must be used instead. When this
# property is set then those two properties are ignored.
#
# Example,
#   ocsp.responderCertSubjectName="CN=OCSP Responder, O=XYZ Corp"

#
# Issuer name of the OCSP responder's certificate
#
# By default, the certificate of the OCSP responder is that of the issuer
# of the certificate being validated. This property identifies the certificate
# of the OCSP responder when the default does not apply. Its value is a string
# distinguished name (defined in RFC 2253) which identifies a certificate in
# the set of certificates supplied during cert path validation. When this
# property is set then the "ocsp.responderCertSerialNumber" property must also
# be set. When the "ocsp.responderCertSubjectName" property is set then this
# property is ignored.
#
# Example,
#   ocsp.responderCertIssuerName="CN=Enterprise CA, O=XYZ Corp"

#
# Serial number of the OCSP responder's certificate
#
# By default, the certificate of the OCSP responder is that of the issuer
# of the certificate being validated. This property identifies the certificate
# of the OCSP responder when the default does not apply. Its value is a string
# of hexadecimal digits (colon or space separators may be present) which
# identifies a certificate in the set of certificates supplied during cert path
# validation. When this property is set then the "ocsp.responderCertIssuerName"
# property must also be set. When the "ocsp.responderCertSubjectName" property
# is set then this property is ignored.
#
# Example,
#   ocsp.responderCertSerialNumber=2A:FF:00

#
# Policy for failed Kerberos KDC lookups:
#
# When a KDC is unavailable (network error, service failure, etc), it is
# put inside a blacklist and accessed less often for future requests. The
# value (case-insensitive) for this policy can be:
#
# tryLast
#    KDCs in the blacklist are always tried after those not on the list.
#
# tryLess[:max_retries,timeout]
#    KDCs in the blacklist are still tried by their order in the configuration,
#    but with smaller max_retries and timeout values. max_retries and timeout
#    are optional numerical parameters (default 1 and 5000, which means once
#    and 5 seconds). Please notes that if any of the values defined here is
#    more than what is defined in krb5.conf, it will be ignored.
#
# Whenever a KDC is detected as available, it is removed from the blacklist.
# The blacklist is reset when krb5.conf is reloaded. You can add
# refreshKrb5Config=true to a JAAS configuration file so that krb5.conf is
# reloaded whenever a JAAS authentication is attempted.
#
# Example,
#   krb5.kdc.bad.policy = tryLast
#   krb5.kdc.bad.policy = tryLess:2,2000
krb5.kdc.bad.policy = tryLast

#
# This property contains a list of disabled EC Named Curves that can be included
# in the jdk.[tls|certpath|jar].disabledAlgorithms properties.  To include this
# list in any of the disabledAlgorithms properties, add the property name as
# an entry.
jdk.disabled.namedCurves = secp112r1, secp112r2, secp128r1, secp128r2, \
    secp160k1, secp160r1, secp160r2, secp192k1, secp192r1, secp224k1, \
    secp224r1, secp256k1, sect113r1, sect113r2, sect131r1, sect131r2, \
    sect163k1, sect163r1, sect163r2, sect193r1, sect193r2, sect233k1, \
    sect233r1, sect239k1, sect283k1, sect283r1, sect409k1, sect409r1, \
    sect571k1, sect571r1, X9.62 c2tnb191v1, X9.62 c2tnb191v2, \
    X9.62 c2tnb191v3, X9.62 c2tnb239v1, X9.62 c2tnb239v2, X9.62 c2tnb239v3, \
    X9.62 c2tnb359v1, X9.62 c2tnb431r1, X9.62 prime192v2, X9.62 prime192v3, \
    X9.62 prime239v1, X9.62 prime239v2, X9.62 prime239v3, brainpoolP256r1, \
    brainpoolP320r1, brainpoolP384r1, brainpoolP512r1

# Algorithm restrictions for certification path (CertPath) processing
#
# In some environments, certain algorithms or key lengths may be undesirable
# for certification path building and validation.  For example, "MD2" is
# generally no longer considered to be a secure hash algorithm.  This section
# describes the mechanism for disabling algorithms based on algorithm name
# and/or key length.  This includes algorithms used in certificates, as well
# as revocation information such as CRLs and signed OCSP Responses.
# The syntax of the disabled algorithm string is described as follows:
#   DisabledAlgorithms:
#       " DisabledAlgorithm { , DisabledAlgorithm } "
#
#   DisabledAlgorithm:
#       AlgorithmName [Constraint] { '&' Constraint } | IncludeProperty
#
#   AlgorithmName:
#       (see below)
#
#   Constraint:
#       KeySizeConstraint | CAConstraint | DenyAfterConstraint |
#       UsageConstraint
#
#   KeySizeConstraint:
#       keySize Operator KeyLength
#
```

```
#   Operator:
#       <= | < | == | != | >= | >
#
#   KeyLength:
#       Integer value of the algorithm's key length in bits
#
#   CAConstraint:
#       jdkCA
#
#   DenyAfterConstraint:
#       denyAfter YYYY-MM-DD
#
#   UsageConstraint:
#       usage [TLSServer] [TLSClient] [SignedJAR]
#
#   IncludeProperty:
#       include <security property>
#
# The "AlgorithmName" is the standard algorithm name of the disabled
# algorithm. See "Java Cryptography Architecture Standard Algorithm Name
# Documentation" for information about Standard Algorithm Names.  Matching
# is performed using a case-insensitive sub-element matching rule.  (For
# example, in "SHA1withECDSA" the sub-elements are "SHA1" for hashing and
# "ECDSA" for signatures.)  If the assertion "AlgorithmName" is a
# sub-element of the certificate algorithm name, the algorithm will be
# rejected during certification path building and validation.  For example,
# the assertion algorithm name "DSA" will disable all certificate algorithms
# that rely on DSA, such as NONEwithDSA, SHA1withDSA.  However, the assertion
# will not disable algorithms related to "ECDSA".
#
# The "IncludeProperty" allows a implementation-defined security property that
# can be included in the disabledAlgorithms properties.  These properties are
# to help manage common actions easier across multiple disabledAlgorithm
# properties.
# There is one defined security property:  jdk.disabled.NamedCurves
# See the property for more specific details.
#
#
# A "Constraint" defines restrictions on the keys and/or certificates for
# a specified AlgorithmName:
#
#   KeySizeConstraint:
#     keySize Operator KeyLength
#       The constraint requires a key of a valid size range if the
#       "AlgorithmName" is of a key algorithm.  The "KeyLength" indicates
#       the key size specified in number of bits.  For example,
#       "RSA keySize <= 1024" indicates that any RSA key with key size less
#       than or equal to 1024 bits should be disabled, and
#       "RSA keySize < 1024, RSA keySize > 2048" indicates that any RSA key
#       with key size less than 1024 or greater than 2048 should be disabled.
#       This constraint is only used on algorithms that have a key size.
#
#   CAConstraint:
#     jdkCA
#       This constraint prohibits the specified algorithm only if the
#       algorithm is used in a certificate chain that terminates at a marked
#       trust anchor in the lib/security/cacerts keystore.  If the jdkCA
#       constraint is not set, then all chains using the specified algorithm
#       are restricted.  jdkCA may only be used once in a DisabledAlgorithm
#       expression.
#       Example:  To apply this constraint to SHA-1 certificates, include
#       the following:  "SHA1 jdkCA"
#
#   DenyAfterConstraint:
#     denyAfter YYYY-MM-DD
#       This constraint prohibits a certificate with the specified algorithm
#       from being used after the date regardless of the certificate's
#       validity.  JAR files that are signed and timestamped before the
#       constraint date with certificates containing the disabled algorithm
#       will not be restricted.  The date is processed in the UTC timezone.
#       This constraint can only be used once in a DisabledAlgorithm
#       expression.
#       Example:  To deny usage of RSA 2048 bit certificates after Feb 3 2020,
#       use the following:  "RSA keySize == 2048 & denyAfter 2020-02-03"
#
#   UsageConstraint:
#     usage [TLSServer] [TLSClient] [SignedJAR]
#       This constraint prohibits the specified algorithm for
#       a specified usage.  This should be used when disabling an algorithm
#       for all usages is not practical. 'TLSServer' restricts the algorithm
#       in TLS server certificate chains when server authentication is
#       performed. 'TLSClient' restricts the algorithm in TLS client
#       certificate chains when client authentication is performed.
#       'SignedJAR' constrains use of certificates in signed jar files.
#       The usage type follows the keyword and more than one usage type can
#       be specified with a whitespace delimiter.
#       Example:  "SHA1 usage TLSServer TLSClient"
#
```

```
# When an algorithm must satisfy more than one constraint, it must be
# delimited by an ampersand '&'.  For example, to restrict certificates in a
# chain that terminate at a distribution provided trust anchor and contain
# RSA keys that are less than or equal to 1024 bits, add the following
# constraint:  "RSA keySize <= 1024 & jdkCA".
#
# All DisabledAlgorithms expressions are processed in the order defined in the
# property.  This requires lower keysize constraints to be specified
# before larger keysize constraints of the same algorithm.  For example:
# "RSA keySize < 1024 & jdkCA, RSA keySize < 2048".
#
# Note: The algorithm restrictions do not apply to trust anchors or
# self-signed certificates.
#
# Note: This property is currently used by Oracle's PKIX implementation. It
# is not guaranteed to be examined and used by other implementations.
#
# Example:
#   jdk.certpath.disabledAlgorithms=MD2, DSA, RSA keySize < 2048
#
#
jdk.certpath.disabledAlgorithms=MD2, MD5, SHA1 jdkCA & usage TLSServer, \
    RSA keySize < 1024, DSA keySize < 1024, EC keySize < 224, \
    include jdk.disabled.namedCurves

#
# Legacy algorithms for certification path (CertPath) processing and
# signed JAR files.
#
# In some environments, a certain algorithm or key length may be undesirable
# but is not yet disabled.
#
# Tools such as keytool and jarsigner may emit warnings when these legacy
# algorithms are used. See the man pages for those tools for more information.
#
# The syntax is the same as the "jdk.certpath.disabledAlgorithms" and
# "jdk.jar.disabledAlgorithms" security properties.
#
# Note: This property is currently used by the JDK Reference
# implementation. It is not guaranteed to be examined and used by other
# implementations.

jdk.security.legacyAlgorithms=SHA1, \
    RSA keySize < 2048, DSA keySize < 2048
```

```
#
# Algorithm restrictions for signed JAR files
#
# In some environments, certain algorithms or key lengths may be undesirable
# for signed JAR validation.  For example, "MD2" is generally no longer
# considered to be a secure hash algorithm.  This section describes the
# mechanism for disabling algorithms based on algorithm name and/or key length.
# JARs signed with any of the disabled algorithms or key sizes will be treated
# as unsigned.
#
# The syntax of the disabled algorithm string is described as follows:
#   DisabledAlgorithms:
#       " DisabledAlgorithm { , DisabledAlgorithm } "
#
#   DisabledAlgorithm:
#       AlgorithmName [Constraint] { '&' Constraint }
#
#   AlgorithmName:
#       (see below)
#
#   Constraint:
#       KeySizeConstraint | DenyAfterConstraint
#
#   KeySizeConstraint:
#       keySize Operator KeyLength
#
#   DenyAfterConstraint:
#       denyAfter YYYY-MM-DD
#
#   Operator:
#       <= | < | == | != | >= | >
#
#   KeyLength:
#       Integer value of the algorithm's key length in bits
#
# Note: This property is currently used by the JDK Reference
# implementation. It is not guaranteed to be examined and used by other
# implementations.
#
# See "jdk.certpath.disabledAlgorithms" for syntax descriptions.
#
jdk.jar.disabledAlgorithms=MD2, MD5, RSA keySize < 1024, \
      DSA keySize < 1024, include jdk.disabled.namedCurves
```

```
#
# Algorithm restrictions for Secure Socket Layer/Transport Layer Security
# (SSL/TLS) processing
#
# In some environments, certain algorithms or key lengths may be undesirable
# when using SSL/TLS.  This section describes the mechanism for disabling
# algorithms during SSL/TLS security parameters negotiation, including
# protocol version negotiation, cipher suites selection, signature schemes
# selection, peer authentication and key exchange mechanisms.
#
# Disabled algorithms will not be negotiated for SSL/TLS connections, even
# if they are enabled explicitly in an application.
#
# For PKI-based peer authentication and key exchange mechanisms, this list
# of disabled algorithms will also be checked during certification path
# building and validation, including algorithms used in certificates, as
# well as revocation information such as CRLs and signed OCSP Responses.
# This is in addition to the jdk.certpath.disabledAlgorithms property above.
#
# See the specification of "jdk.certpath.disabledAlgorithms" for the
# syntax of the disabled algorithm string.
#
# Note: The algorithm restrictions do not apply to trust anchors or
# self-signed certificates.
#
# Note: This property is currently used by the JDK Reference implementation.
# It is not guaranteed to be examined and used by other implementations.
#
# Example:
#   jdk.tls.disabledAlgorithms=MD5, SSLv3, DSA, RSA keySize < 2048, \
#       rsa_pkcs1_sha1
jdk.tls.disabledAlgorithms=SSLv3, TLSv1, TLSv1.1, RC4, DES, MD5withRSA, DH keySize <
1024, DESede, \
     EC keySize < 224, 3DES_EDE_CBC, anon, NULL, DES_CBC, \
   include jdk.disabled.namedCurves



# Legacy algorithms for Secure Socket Layer/Transport Layer Security (SSL/TLS)
# processing in JSSE implementation.
#
# In some environments, a certain algorithm may be undesirable but it
# cannot be disabled because of its use in legacy applications.  Legacy
# algorithms may still be supported, but applications should not use them
# as the security strength of legacy algorithms are usually not strong enough
# in practice.
#
# During SSL/TLS security parameters negotiation, legacy algorithms will
# not be negotiated unless there are no other candidates.
#
# The syntax of the legacy algorithms string is described as this Java
# BNF-style:
#   LegacyAlgorithms:
#       " LegacyAlgorithm { , LegacyAlgorithm } "
#
#   LegacyAlgorithm:
#       AlgorithmName (standard JSSE algorithm name)
#
# See the specification of security property "jdk.certpath.disabledAlgorithms"
# for the syntax and description of the "AlgorithmName" notation.
#
# Cipher suites have the form:
#       SSL_KeyExchangeAlg_WITH_CipherAlg_MacAlg
#
# For example, the cipher suite SSL_RSA_WITH_AES_128_CBC_SHA uses RSA as the
# key exchange algorithm, AES_128_CBC (128 bits AES cipher algorithm in CBC
# mode) as the cipher (encryption) algorithm, and SHA-1 as the message digest
# algorithm for HMAC.
#
# The LegacyAlgorithm can be one of the following standard algorithm names:
#     1. JSSE cipher suite name, e.g., SSL_RSA_WITH_AES_128_CBC_SHA
#     2. JSSE key exchange algorithm name, e.g., RSA
#     3. JSSE cipher (encryption) algorithm name, e.g., AES_128_CBC
#     4. JSSE message digest algorithm name, e.g., SHA
#
# See SSL/TLS specifications and "Java Cryptography Architecture Standard
# Algorithm Name Documentation" for information about the algorithm names.
#
# Note: This property is currently used by the JDK Reference implementation.
# It is not guaranteed to be examined and used by other implementations.
# There is no guarantee the property will continue to exist or be of the
# same syntax in future releases.
#
# Example:
#   jdk.tls.legacyAlgorithms=DH_anon, DES_CBC, SSL_RSA_WITH_RC4_128_MD5
#
jdk.tls.legacyAlgorithms= \
        K_NULL, C_NULL, M_NULL, \
        DH_anon, ECDH_anon, \
        RC4_128, RC4_40, DES_CBC, DES40_CBC

# The pre-defined default finite field Diffie-Hellman ephemeral (DHE)
# parameters for Transport Layer Security (SSL/TLS/DTLS) processing.
#
# In traditional SSL/TLS/DTLS connections where finite field DHE parameters
# negotiation mechanism is not used, the server offers the client group
# parameters, base generator g and prime modulus p, for DHE key exchange.
# It is recommended to use dynamic group parameters.  This property defines
# a mechanism that allows you to specify custom group parameters.
#
# The syntax of this property string is described as this Java BNF-style:
#   DefaultDHEParameters:
#       DefinedDHEParameters { , DefinedDHEParameters }
#
#   DefinedDHEParameters:
#       "{" DHEPrimeModulus , DHEBaseGenerator "}"
#
#   DHEPrimeModulus:
#       HexadecimalDigits
#
#   DHEBaseGenerator:
#       HexadecimalDigits
#
#   HexadecimalDigits:
#       HexadecimalDigit { HexadecimalDigit }
#
#   HexadecimalDigit: one of
#       0 1 2 3 4 5 6 7 8 9 A B C D E F a b c d e f
#
# Whitespace characters are ignored.
#
# The "DefinedDHEParameters" defines the custom group parameters, prime
# modulus p and base generator g, for a particular size of prime modulus p.
# The "DHEPrimeModulus" defines the hexadecimal prime modulus p, and the
# "DHEBaseGenerator" defines the hexadecimal base generator g of a group
# parameter.  It is recommended to use safe primes for the custom group
# parameters.
#
# If this property is not defined or the value is empty, the underlying JSSE
# provider's default group parameter is used for each connection.
#
```

```
# If the property value does not follow the grammar, or a particular group
# parameter is not valid, the connection will fall back and use the
# underlying JSSE provider's default group parameter.
#
# Note: This property is currently used by OpenJDK's JSSE implementation. It
# is not guaranteed to be examined and used by other implementations.
#
# Example:
#   jdk.tls.server.defaultDHEParameters=
#       { \
#       FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1 \
#       29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD \
#       EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245 \
#       E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED \
#       EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE65381 \
#       FFFFFFFF FFFFFFFF, 2}

# Cryptographic Jurisdiction Policy defaults
#
# Import and export control rules on cryptographic software vary from
# country to country.  By default, the JDK provides two different sets of
# cryptographic policy files:
#
#     unlimited:  These policy files contain no restrictions on cryptographic
#                 strengths or algorithms.
#
#     limited:    These policy files contain more restricted cryptographic
#                 strengths, and are still available if your country or
#                 usage requires the traditional restrictive policy.
#
# The JDK JCE framework uses the unlimited policy files by default.
# However the user may explicitly choose a set either by defining the
# "crypto.policy" Security property or by installing valid JCE policy
# jar files into the traditional JDK installation location.  To better
# support older JDK Update releases, the "crypto.policy" property is not
# defined by default.  See below for more information.
#
# The following logic determines which policy files are used:
#
#         <java-home> refers to the directory where the JRE was
#         installed and may be determined using the "java.home"
#         System property.
#
# 1.  If the Security property "crypto.policy" has been defined,
#     then the following mechanism is used:
#
#     The policy files are stored as jar files in subdirectories of
# <java-home>/lib/security/policy.  Each directory contains a complete
# set of policy files.
#
#     The "crypto.policy" Security property controls the directory
#     selection, and thus the effective cryptographic policy.
#
# The default set of directories is:
#
#     limited | unlimited
#
# 2.  If the "crypto.policy" property is not set and the traditional
#     US_export_policy.jar and local_policy.jar files
#     (e.g. limited/unlimited) are found in the legacy
#     <java-home>/lib/security directory, then the rules embedded within
#     those jar files will be used. This helps preserve compatibility
# for users upgrading from an older installation.
#
# 3.  If the jar files are not present in the legacy location
#     and the "crypto.policy" Security property is not defined,
#     then the JDK will use the unlimited settings (equivalent to
#     crypto.policy=unlimited)
#
# Please see the JCA documentation for additional information on these
# files and formats.
#
# YOU ARE ADVISED TO CONSULT YOUR EXPORT/IMPORT CONTROL COUNSEL OR
ATTORNEY
# TO DETERMINE THE EXACT REQUIREMENTS.
#
# Please note that the JCE for Java SE, including the JCE framework,
# cryptographic policy files, and standard JCE providers provided with
# the Java SE, have been reviewed and approved for export as mass market
# encryption item by the US Bureau of Industry and Security.
#
# Note: This property is currently used by the JDK Reference implementation.
# It is not guaranteed to be examined and used by other implementations.
#
#crypto.policy=unlimited

# CORBA ORBIorTypeCheckRegistryFilter
# Type check enhancement for ORB::string_to_object processing
```

```
#
# An IOR type check filter, if configured, is used by an ORB during
# an ORB::string_to_object invocation to check the veracity of the type encoded
# in the ior string.
#
# The filter pattern consists of a semi-colon separated list of class names.
# The configured list contains the binary class names of the IDL interface types
# corresponding to the IDL stub class to be instantiated.
# As such, a filter specifies a list of IDL stub classes that will be
# allowed by an ORB when an ORB::string_to_object is invoked.
# It is used to specify a white list configuration of acceptable
# IDL stub types which may be contained in a stringified IOR
# parameter passed as input to an ORB::string_to_object method.
#
# Note: This property is currently used by the JDK Reference implementation.
# It is not guaranteed to be examined and used by other implementations.
#
#com.sun.CORBA.ORBIorTypeCheckRegistryFilter=binary_class_name;binary_class_name

#
# JCEKS Encrypted Key Serial Filter
#
# This filter, if configured, is used by the JCEKS KeyStore during the
# deserialization of the encrypted Key object stored inside a key entry.
# If not configured or the filter result is UNDECIDED (i.e. none of the patterns
# matches), the filter configured by jdk.serialFilter will be consulted.
#
# If the system property jceks.key.serialFilter is also specified, it supersedes
# the security property value defined here.
#
# The filter pattern uses the same format as jdk.serialFilter. The default
# pattern allows javax.crypto.spec.SecretKeySpec, javax.security.auth.kerberos.*,
# com.ibm.crypto.provider.*, com.ibm.crypto.fips.provider.*, com.ibm.crypto.plus.provider.*,
# com.ibm.security.x509.*, java.math.BigInteger;,
# and java.lang.Number and rejects all the others.
jceks.key.serialFilter = javax.crypto.spec.SecretKeySpec;\
  javax.security.auth.kerberos.*;com.ibm.crypto.provider.*;\
  com.ibm.crypto.fips.provider.*;com.ibm.crypto.plus.provider.*;\
  com.ibm.security.x509.*;java.math.BigInteger;java.lang.Number;\
  java.lang.Enum;java.security.KeyRep;java.security.KeyRep$Type;!*
#
# Disabled mechanisms for the Simple Authentication and Security Layer (SASL)
#
# Disabled mechanisms will not be negotiated by both SASL clients and servers.
# These mechanisms will be ignored if they are specified in the "mechanisms"
# argument of "Sasl.createSaslClient" or the "mechanism" argument of
# "Sasl.createSaslServer".
#
# The value of this property is a comma-separated list of SASL mechanisms.
# The mechanisms are case-sensitive. Whitespaces around the commas are ignored.
#
# Note: This property is currently used by the JDK Reference implementation.
# It is not guaranteed to be examined and used by other implementations.
#
# Example:
#   jdk.sasl.disabledMechanisms=PLAIN, CRAM-MD5, DIGEST-MD5
jdk.sasl.disabledMechanisms=

#
# Policies for distrusting Certificate Authorities (CAs).
#
# This is a comma separated value of one or more case-sensitive strings, each
# of which represents a policy for determining if a CA should be distrusted.
# The supported values are:
#
#   SYMANTEC_TLS : Distrust TLS Server certificates anchored by a Symantec
#   root CA and issued after April 16, 2019 unless issued by one of the
#   following subordinate CAs which have a later distrust date:
#     1. Apple IST CA 2 - G1, SHA-256 fingerprint:
#        AC2B922ECFD5E01711772FEA8ED372DE9D1E2245FCE3F57A9CDBEC77296A424B
#        Distrust after December 31, 2019.
#     2. Apple IST CA 8 - G1, SHA-256 fingerprint:
#        A4FE7C7F15155F3F0AEF7AAA83CF6E06DEB97CA3F909DF920AC1490882D488ED
#        Distrust after December 31, 2019.
#
# Leading and trailing whitespace surrounding each value are ignored.
# Unknown values are ignored. If the property is commented out or set to the
# empty String, no policies are enforced.
#
# Note: This property is currently used by the JDK Reference implementation.
# It is not guaranteed to be supported by other SE implementations. Also, this
# property does not override other security properties which can restrict
# certificates such as jdk.tls.disabledAlgorithms or
# jdk.certpath.disabledAlgorithms; those restrictions are still enforced even
# if this property is not enabled.
#
jdk.security.caDistrustPolicies=SYMANTEC_TLS
```

```
#
# Policies for the proxy_impersonator Kerberos ccache configuration entry
#
# The proxy_impersonator ccache configuration entry indicates that the ccache
# is a synthetic delegated credential for use with S4U2Proxy by an intermediate
# server. The ccache file should also contain the TGT of this server and
# an evidence ticket from the default principal of the ccache to this server.
#
# This security property determines how Java uses this configuration entry.
# There are 3 possible values:
#
#  no-impersonate     - Ignore this configuration entry, and always act as
#                       the owner of the TGT (if it exists).
#
#  try-impersonate    - Try impersonation when this configuration entry exists.
#                       If no matching TGT or evidence ticket is found,
#                       fallback to no-impersonate.
#
#  always-impersonate - Always impersonate when this configuration entry exists.
#                       If no matching TGT or evidence ticket is found,
#                       no initial credential is read from the ccache.
#
# The default value is "always-impersonate".
#
# If a system property of the same name is also specified, it supersedes the
# security property value defined here.
#
#jdk.security.krb5.default.initiate.credential=always-impersonate

#
# JNDI Object Factories Filter
#
# This filter is used by the JNDI runtime to control the set of object factory classes
# which will be allowed to instantiate objects from object references returned by
# naming/directory systems. The factory class named by the reference instance will be
# matched against this filter. The filter property supports pattern-based filter syntax
# with the same format as jdk.serialFilter.
#
# Each pattern is matched against the factory class name to allow or disallow it's
# instantiation. The access to a factory class is allowed unless the filter returns
# REJECTED.
#
# Note: This property is currently used by the JDK Reference implementation.
# It is not guaranteed to be examined and used by other implementations.
```

```
#
# If the system property jdk.jndi.object.factoriesFilter is also specified, it supersedes
# the security property value defined here. The default value of the property is "*".
#
# The default pattern value allows any object factory class specified by the reference
# instance to recreate the referenced object.
#jdk.jndi.object.factoriesFilter=*

#
# The default Character set name (java.nio.charset.Charset.forName())
# for converting TLS ALPN values between byte arrays and Strings.
# Prior versions of the JDK may use UTF-8 as the default charset. If
# you experience interoperability issues, setting this property to UTF-8
# may help.
#
# jdk.tls.alpnCharset=UTF-8
jdk.tls.alpnCharset=ISO_8859_1
#
# The policy for the XML Signature secure validation mode. The mode is
# enabled by setting the property "org.jcp.xml.dsig.secureValidation" to
# true with the javax.xml.crypto.XMLCryptoContext.setProperty() method,
# or by running the code with a SecurityManager.
#
#   Policy:
#       Constraint {"," Constraint }
#   Constraint:
#       AlgConstraint | MaxTransformsConstraint | MaxReferencesConstraint |
#       ReferenceUriSchemeConstraint | KeySizeConstraint | OtherConstraint
#   AlgConstraint
#       "disallowAlg" Uri
#   MaxTransformsConstraint:
#       "maxTransforms" Integer
#   MaxReferencesConstraint:
#       "maxReferences" Integer
#   ReferenceUriSchemeConstraint:
#       "disallowReferenceUriSchemes" String { String }
#   KeySizeConstraint:
#       "minKeySize" KeyAlg Integer
#   OtherConstraint:
#       "noDuplicateIds" | "noRetrievalMethodLoops"
#
# For AlgConstraint, Uri is the algorithm URI String that is not allowed.
# See the XML Signature Recommendation for more information on algorithm
# URI Identifiers. For KeySizeConstraint, KeyAlg is the standard algorithm
# name of the key type (ex: "RSA"). If the MaxTransformsConstraint,
# MaxReferencesConstraint or KeySizeConstraint (for the same key type) is
# specified more than once, only the last entry is enforced.
#
# Note: This property is currently used by the JDK Reference implementation. It
# is not guaranteed to be examined and used by other implementations.
#
jdk.xml.dsig.secureValidationPolicy=\
    disallowAlg http://www.w3.org/TR/1999/REC-xslt-19991116,\
    disallowAlg http://www.w3.org/2001/04/xmldsig-more#rsa-md5,\
    disallowAlg http://www.w3.org/2001/04/xmldsig-more#hmac-md5,\
    disallowAlg http://www.w3.org/2001/04/xmldsig-more#md5,\
    maxTransforms 5,\
    maxReferences 30,\
    disallowReferenceUriSchemes file http https,\
    minKeySize RSA 1024,\
    minKeySize DSA 1024,\
    minKeySize EC 224,\
    noDuplicateIds,\
    noRetrievalMethodLoops

#
# Deserialization system-wide filter factory
#
# A filter factory class name is used to configure the system-wide filter factory.
# The filter factory selects the sun.misc.ObjectInputFilter to use for each
# ObjectInputStream when invoked with a current and a requested filter.
# The class must be public, must have a public zero-argument constructor, implement the
# java.util.function.BinaryOperator<sun.misc.ObjectInputFilter> interface,
# provide its implementation and be accessible via the application class loader.
# See the release notes for more details.
#
# If the system property jdk.serialFilterFactory is also specified, it supersedes
# the security property value defined here.
#
#jdk.serialFilterFactory=<classname>

#
# Serialization process-wide filter
#
# A filter, if configured, is used by java.io.ObjectInputStream during
# deserialization to check the contents of the stream.
# A filter is configured as a sequence of patterns, each pattern is either
# matched against the name of a class in the stream or defines a limit.
# Patterns are separated by ";" (semicolon).
# Whitespace is significant and is considered part of the pattern.
#
# If the system property jdk.serialFilter is also specified on the command
# line, it supersedes the security property value defined here.
#
# If a pattern includes a "=", it sets a limit.
# If a limit appears more than once the last value is used.
# Limits are checked before classes regardless of the order in the sequence of patterns.
# If any of the limits are exceeded, the filter status is REJECTED.
#
#   maxdepth=value - the maximum depth of a graph
#   maxrefs=value  - the maximum number of internal references
#   maxbytes=value - the maximum number of bytes in the input stream
#   maxarray=value - the maximum array length allowed
#
# Other patterns, from left to right, match the class or package name as
# returned from Class.getName.
# If the class is an array type, the class or package to be matched is the element type.
# Arrays of any number of dimensions are treated the same as the element type.
# For example, a pattern of "!example.Foo", rejects creation of any instance or
# array of example.Foo.
#
# If the pattern starts with "!", the status is REJECTED if the remaining pattern
#   is matched; otherwise the status is ALLOWED if the pattern matches.
# If the pattern ends with ".**" it matches any class in the package and all subpackages.
# If the pattern ends with ".*" it matches any class in the package.
# If the pattern ends with "*", it matches any class with the pattern as a prefix.
# If the pattern is equal to the class name, it matches.
# Otherwise, the status is UNDECIDED.
#
# Primitive types are not configurable with this filter.
#
#jdk.serialFilter=pattern;pattern

#
# RMI Registry Serial Filter
#
# The filter pattern uses the same format as jdk.serialFilter.
# This filter can override the builtin filter if additional types need to be
# allowed or rejected from the RMI Registry or to decrease limits but not
# to increase limits.
# If the limits (maxdepth, maxrefs, or maxbytes) are exceeded, the object is rejected.
#
```

```
# The maxdepth of any array passed to the RMI Registry is set to
# 10000.  The maximum depth of the graph is set to 20.
# These limits can be reduced via the maxarray, maxdepth limits.
#
sun.rmi.registry.registryFilter=javax.rmi.CORBA.Stub
# Each non-array type is allowed or rejected if it matches one of the patterns,
# evaluated from left to right, and is otherwise allowed. Arrays of any
# component type, including subarrays and arrays of primitives, are allowed.
#
# Array construction of any component type, including subarrays and arrays of
# primitives, are allowed unless the length is greater than the maxarray limit.
# The filter is applied to each array element.
#
# The built-in filter allows subclasses of allowed classes and
# can approximately be represented as the pattern:
#
#sun.rmi.registry.registryFilter=\
#    maxarray=1000000;\
#    maxdepth=20;\
#    java.lang.String;\
#    java.lang.Number;\
#    java.lang.reflect.Proxy;\
#    java.rmi.Remote;\
#    sun.rmi.server.UnicastRef;\
#    sun.rmi.server.RMIClientSocketFactory;\
#    sun.rmi.server.RMIServerSocketFactory;\
#    java.rmi.activation.ActivationID;\
#    java.rmi.server.UID
#
# RMI Distributed Garbage Collector (DGC) Serial Filter
#
# The filter pattern uses the same format as jdk.serialFilter.
# This filter can override the builtin filter if additional types need to be
# allowed or rejected from the RMI DGC.
#
# The builtin DGC filter can approximately be represented as the filter pattern:
#
#sun.rmi.transport.dgcFilter=\
#    java.rmi.server.ObjID;\
#    java.rmi.server.UID;\
#    java.rmi.dgc.VMID;\
#    java.rmi.dgc.Lease;\
#    maxdepth=5;maxarray=10000
```

```
#
# TLS key limits on symmetric cryptographic algorithms
#
# This security property sets limits on algorithms key usage in TLS 1.3.
# When the amount of data encrypted exceeds the algorithm value listed below,
# a KeyUpdate message will trigger a key change.  This is for symmetric ciphers
# with TLS 1.3 only.
#
# The syntax for the property is described below:
#   KeyLimits:
#       " KeyLimit { , KeyLimit } "
#
#   WeakKeyLimit:
#       AlgorithmName Action Length
#
#   AlgorithmName:
#       A full algorithm transformation.
#
#   Action:
#       KeyUpdate
#
#   Length:
#       The amount of encrypted data in a session before the Action occurs
#       This value may be an integer value in bytes, or as a power of two, 2^29.
#
#   KeyUpdate:
#       The TLS 1.3 KeyUpdate handshake process begins when the Length amount
#       is fulfilled.
#
# Note: This property is currently used by OpenJDK's JSSE implementation. It
# is not guaranteed to be examined and used by other implementations.
#
jdk.tls.keyLimits=AES/GCM/NoPadding KeyUpdate 2^37
```