DoD Public Key Enablement (PKE) Reference Guide

PKI Interoperability Test Tool version 2.0.6 (PITTv2) User Guide

Contact: dodpke@mail.mil
URL: http://iase.disa.mil/pki-pke

Enabling PKI Technology for DoD users

PKI Interoperability Test Tool v2.0.6 (PITTv2) User Guide

September 2, 2015

Version 2.0.6

DoD PKE Team

Revision History

Issue Date	Revision	Change Description
06/20/2014	2.0.0	Initial release
01/15/15	2.0.5	Updated Linux installation instructions
09/02/15	2.0.6	Updated screenshots to reflect interface changes and provided descriptions for the new checkboxes.

Contents

Overview

Background

The PKI Interoperability Test Tool (PITT) was developed in 2008 to assist with evaluating interoperability alternatives to establish trust with prospective partner PKIs and to troubleshoot certification path processing problems. The tool was maintained until Fall 2010, when version 1.2 was released. Since that time, the libraries used by PITT have not been maintained, complicating debugging and enhancement of PITT.

PITTv2 is a re-write of PITT using the libraries that underpin the Trust Anchor Constraints Tool (TACT) and Install Root v4 products. This avoids the need to refresh PITT and all dependencies while providing a good test platform for a currently used library. Additionally, the current libraries facilitate some nice features not available in PITT. Some seldom-used features from PITT were dropped in PITTv2, as described in Appendix A.

Intended Audience

This document is intended for those using PITTv2 to troubleshoot certification path processing issues or to conduct interoperability tests.

Supplemental Information

The DoD Public Key Enablement (PKE) web site located at http://iase.disa.mil/pki-pke contains many informational documents and best practice guides related to PK-enablement and certificate validation implementation in the DoD. TACT usage guides, are available on the site as well. TACT and PITTv2 use the same configuration file formats and similar user interfaces for configuration.

Installation

Installing PITTv2 on Windows

On Windows platforms, PITTv2 is installed by simply double-clicking the msi to begin installation. The resulting installation wizard contains few choices and is intuitive.

Installing PITTv2 on Linux

On Linux, to install PITTv2 extracting the provided archive and install the rpm package using the rpm utility or yum (e.g. “`yum install pittv2-2.0.1-1.x86_64.el6.rpm`” if required dependencies are missing you will be prompted to install them)

PITTv2 Overview

Configuration Data

PITTv2 relies on several forms of configuration data including certification path validation settings, TACT TA stores and other constraints. The settings are stored in SQLite database files using formats shared with TACT, as described in Appendix B. Most configuration data can be prepared using the Options->Preferences menu item in PITTv2. The TACT TA Store Manager utility is required to manage TACT TA store contents. A set of TA stores containing various collections of DoD-approved trust anchors is installed with PITTv2. The following diagram provides a visual summary of the configuration data.

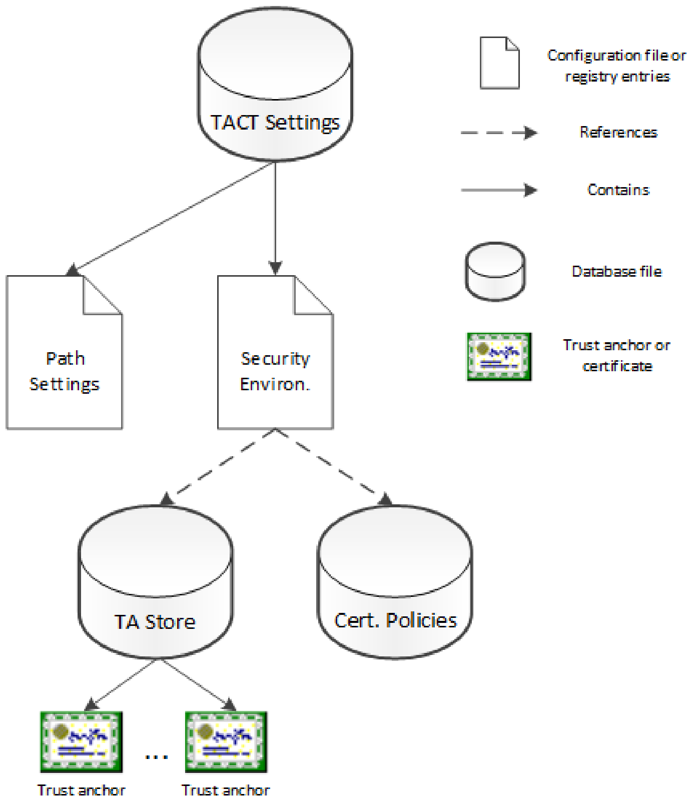


Figure 1 Configuration data

The following subsections describe each type of data used by PITTv2. See the TACT user guide for additional information on these and other related types of data used by TACT.

Logging settings

PITTv2 uses the log4cpp library for logging support, which uses a textual configuration file format from the log4j library. An introduction to the logging configuration file format, as well as library capabilities can be found at: http://logging.apache.org/log4j/1.2/manual.html.

PITTv2 provides tools for editing logging configuration files. While these files can be prepared and maintained manually, it is recommended that the provided tools be used to prepare and maintain logging configuration files. The Edit logging configuration section provides additional information.

TACT Settings

PITTv2 uses the settings file formats used by TACT. TACT settings are stored in a SQLite database and include three different types of configuration items: path settings definitions, security environment definitions and protected resource definitions. PITTv2 does not use protected resource definitions, though such definitions may be present in a settings database file used by PITTv2.

Path settings

Path settings configuration objects contain standard RFC 5280 certification path validation algorithm inputs and other certification path-processing related values. The RFC 5280 configuration items include:

- Initial require explicit policy flag
- Initial inhibit any policy flag
- Initial inhibit policy mapping flag
- Permitted names
- Excluded names
- Initial policy set

Other configuration items include:

- Enforce trust anchor-based constraints (from RFC 5937)
- Enforce cryptographic algorithm and key size constraints (custom flag)

Configuration items related to dynamic certification path discovery include:

- Ignore expired certificates flag
- Retrieve certificates using AIA/SIA extensions – HTTP flag
- Retrieve certificates using AIA/SIA extensions – LDAP flag
- Certificates folder

- Blacklisted certificates

Configuration items related to revocation status determination include:

- Check revocation status flag
- Check OCSP AIA flag
- Check CRLs flag
- Check CRL DP – HTTP flag
- Check CRL DP – LDAP flag
- Only apply CRL grace periods as a last resort flag
- CRL grace period and freshness settings
- Locally trusted OCSP responder settings
- Blacklisted certificates
- Whitelisted (no revocation check) certificates
- CRL folder
- OCSP AIA nonce setting

The Edit path settings section provides additional information on path settings.

Security environment

Security environment configuration items are constraints that apply more broadly than path settings configuration items. For example, algorithm and key size constraints are stored in security environment configuration items, as are trust anchor store references and certificate policy database references. These items tend not to vary from one protected resource definition to the next.

The Edit security environments section provides additional information.

Trust anchor store

The TACT trust anchor store contains RFC 5914 trust anchors. Each trust anchor is an X.509 certificate wrapped in a TrustAnchorChoice structure, with optional constraints defined. Additional information on the structure of TACT trust anchor stores is provided in Appendix B.

Certificate policies database

Certificate policies are stored in a SQLite database and used to simplify the creation of certificate policy-related constraints by allowing a web server administrator to define constraints in terms of policies associated with the enterprise of the administrator, i.e., policies that are familiar to the administrator. The database schema is provided in Appendix B. The use of a certificate policies database is optional.

Application preferences

A variety of application preferences are saved across application invocations. These preferences include window size and placement, last folders used, etc. On Linux, these preferences are saved to a .PITTv2 file in the user's home directory. On Windows, these preferences are saved to the system registry under the following registry hive:

- HKEY_CURRENT_USER\SOFTWARE\Red Hound Software

Application preferences can be safely deleted. New preferences will be established upon the next execution as necessary.

Quick Start Guide

The PITT installer includes similar configuration files as those included with TACT, with a few modifications related to dynamic path discovery and revocation status determination. These configuration files support validating certificates using trust anchors and path validation settings consistent with DoD-approved external PKIs. The default logging configuration saves INFO level and above messages to the system event log on Windows and to the console on Linux. The logging configuration can be adjusted as described in Edit Logging Configuration section.

There are two types of settings that influence certification path processing: path settings and security environments. Path settings include the standard RFC 5280 path validation algorithm inputs, plus a few TACT-specific settings. Security environments reference trust anchor stores that include trust anchors used during path building and validation, as well as algorithm and key size constraints.

Default path settings configurations

The names of the path settings configurations are intended to be intuitive and include the following:

- DoD Medium Hardware 2048 and equivalent partner policies
- DoD Medium Hardware 2048 only
- Default Path Settings

All path settings options require certification paths to be valid under at least one certificate policy. The "Default Path Settings" configuration permits any certificate policy. The "DoD Medium Hardware 2048 only configuration" requires all certification paths to be valid under the DoD Medium Hardware 2048 policy (). The "DoD Medium Hardware 2048 and equivalent partner policies" configuration requires all certification paths to be valid under the DoD Medium Hardware 2048 policy or any policy marked as equivalent in the FedPolicies.pdb file, which was used to create the initial policy set for this configuration.

All path settings options support retrieving certificates from HTTP URIs expressed in AIA and SIA extensions. By default, certificates will be stored in C:\PittSettings\certs on Windows and ~/.PITTv2 on Linux. Expired certificates are ignored during certification path building by default.

All path settings options have revocation status determination turned on with support for using OCSP responders identified in AIA extension, locally available CRLs or CRLs retrieved from HTTP URIs expressed in CRL DP extensions. By default, CRLs will be stored in C:\PittSettings\certs on Windows and ~/.PITTv2 on Linux.

Additional certificates or CRLs can be preplaced in the folders used by PITTv2. These can augment the materials downloaded from remote sources or support offline certification path processing. To enable offline processing, edit the configuration of interest and make sure all settings associated with remote retrieval are not checked, including:

- Retrieve certificates using AIA/SIA extensions - HTTP
- Retrieve certificates using AIA/SIA extensions – LDAP
- Check OCSP AIA
- Check CRL DP – HTTP
- Check CRL DP – LDAP

Make sure the **Certificate folder** and **CRL folder** settings reference the folder where materials are staged and make sure **Check revocation status** and **Check CRLs** are both checked, if revocation status determination is desired.

Default security environment configurations

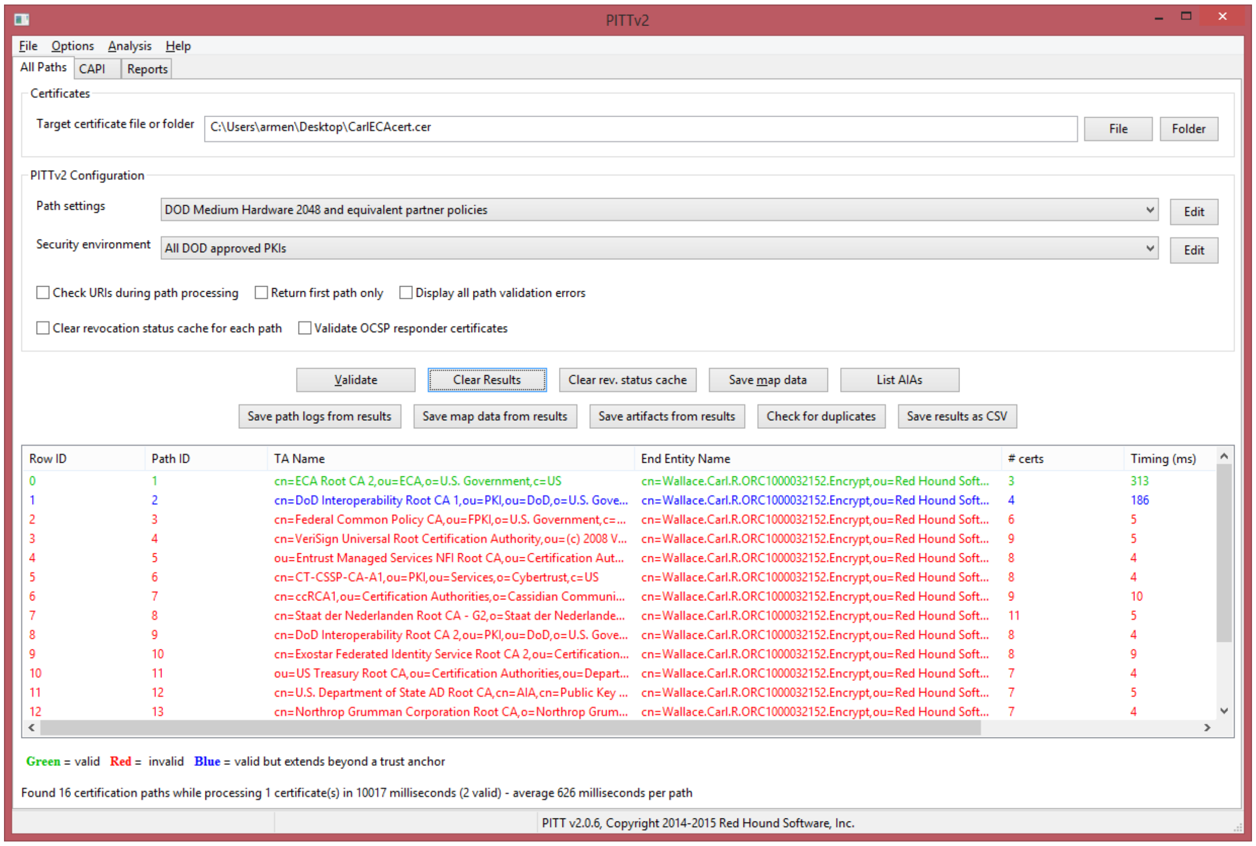
The names of the security environments are intended to be intuitive and include:

- All DOD approved PKIs
- DOD, ECA and Federal
- DOD and ECA
- DOD only

The security environment names provide an indication of the contents of the trust anchor store referenced by the configuration. Trust anchor stores can be edited using the TA Store Manager utility included with TACT.

Using PITTv2

When PITTv2 is started, the All Paths panel will be displayed. To validate a certificate, click the **File** button and browse to a certificate to validate. Select a **Path settings** and a **Security environment** then click **Validate**. In the following example, a certificate issued by an ECA vendor is validated using the "DoD Medium Hardware 2048 and equivalent partner policies" path settings configuration and the "All DOD approved PKIs" security environment.



Certificates

Target certificate file or folder: C:\Users\armen\Desktop\CarlECAcert.cer

PITTv2 Configuration

Path settings: DOD Medium Hardware 2048 and equivalent partner policies

Security environment: All DOD approved PKIs

☐ Check URIs during path processing ☐ Return first path only ☐ Display all path validation errors

☐ Clear revocation status cache for each path ☐ Validate OCSP responder certificates

Row ID	Path ID	TA Name	End Entity Name	# certs	Timing (ms)
0	1	cn=ECA Root CA 2,ou=ECA,o=U.S. Government,c=US	cn=Wallace.Carl.R.ORC1000032152.Encrypt,ou=Red Hound Soft...	3	313
1	2	cn=DoD Interoperability Root CA 1,ou=PKI,ou=DoD,o=U.S. Gove...	cn=Wallace.Carl.R.ORC1000032152.Encrypt,ou=Red Hound Soft...	4	186
2	3	cn=Federal Common Policy CA,ou=FPKI,o=U.S. Government,c=...	cn=Wallace.Carl.R.ORC1000032152.Encrypt,ou=Red Hound Soft...	6	5
3	4	cn=VeriSign Universal Root Certification Authority,ou=(c) 2008 V...	cn=Wallace.Carl.R.ORC1000032152.Encrypt,ou=Red Hound Soft...	9	5
4	5	ou=Entrust Managed Services NFI Root CA,ou=Certification Aut...	cn=Wallace.Carl.R.ORC1000032152.Encrypt,ou=Red Hound Soft...	8	4
5	6	cn=CT-CSSP-CA-A1,ou=PKI,ou=Services,o=Cybertrust,c=US	cn=Wallace.Carl.R.ORC1000032152.Encrypt,ou=Red Hound Soft...	8	4
6	7	cn=ccRCA1,ou=Certification Authorities,o=Cassidian Communi...	cn=Wallace.Carl.R.ORC1000032152.Encrypt,ou=Red Hound Soft...	9	10
7	8	cn=Staat der Nederlanden Root CA - G2,o=Staat der Nederlande...	cn=Wallace.Carl.R.ORC1000032152.Encrypt,ou=Red Hound Soft...	11	5
8	9	cn=DoD Interoperability Root CA 2,ou=PKI,ou=DoD,o=U.S. Gove...	cn=Wallace.Carl.R.ORC1000032152.Encrypt,ou=Red Hound Soft...	8	4
9	10	cn=Exostar Federated Identity Service Root CA 2,ou=Certification...	cn=Wallace.Carl.R.ORC1000032152.Encrypt,ou=Red Hound Soft...	8	9
10	11	ou=US Treasury Root CA,ou=Certification Authorities,ou=Depart...	cn=Wallace.Carl.R.ORC1000032152.Encrypt,ou=Red Hound Soft...	7	4
11	12	cn=U.S. Department of State AD Root CA,cn=AIA,cn=Public Key ...	cn=Wallace.Carl.R.ORC1000032152.Encrypt,ou=Red Hound Soft...	7	5
12	13	cn=Northrop Grumman Corporation Root CA,o=Northrop Grum...	cn=Wallace.Carl.R.ORC1000032152.Encrypt,ou=Red Hound Soft...	7	4

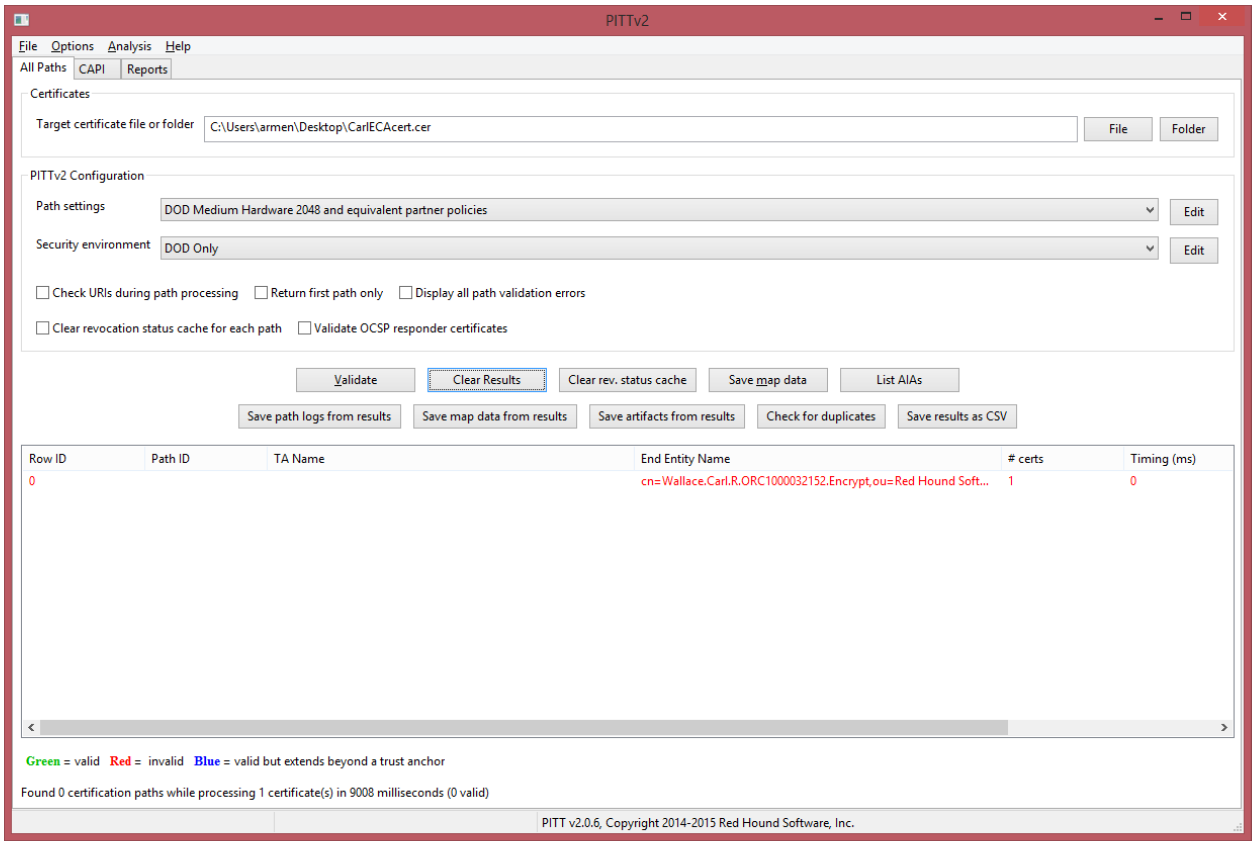
Green = valid **Red** = invalid **Blue** = valid but extends beyond a trust anchor

Found 16 certification paths while processing 1 certificate(s) in 10017 milliseconds (2 valid) - average 626 milliseconds per path

PITT v2.0.6, Copyright 2014-2015 Red Hound Software, Inc.

Figure 2 Validating an ECA certificate with compatible path settings and security environment

Validating the same certificate with different **Path settings** or **Security environment** configurations may yield different results, as shown below when using the "DOD only" security environment. In this case, no paths could be found because the trust anchor store only contains a DOD trust anchor. The bulk of the time spent determining no path is available was spent inspecting URIs present in the AIA extension, as shown in the following log snip from the URI Processing log category. The path settings can be altered to turn AIA chasing to force building against a static, locally available set of certificates. In some environments, it may be necessary to add servers to the blacklist in the security environment as well to avoid unnecessary timeouts.



Certificates

Target certificate file or folder: C:\Users\armen\Desktop\CarlECAcert.cer

PITTv2 Configuration

Path settings: DOD Medium Hardware 2048 and equivalent partner policies

Security environment: DOD Only

☐ Check URIs during path processing ☐ Return first path only ☐ Display all path validation errors

☐ Clear revocation status cache for each path ☐ Validate OCSP responder certificates

Row ID	Path ID	TA Name	End Entity Name	# certs	Timing (ms)
0			cn=Wallace.Carl.R.ORC1000032152.Encrypt,ou=Red Hound Soft...	1	0

Green = valid **Red** = invalid **Blue** = valid but extends beyond a trust anchor

Found 0 certification paths while processing 1 certificate(s) in 9008 milliseconds (0 valid)

PITT v2.0.6, Copyright 2014-2015 Red Hound Software, Inc.

Figure 3 Validating an ECA certificate with DOD only security environment

Menus

File

Clean-up

The Clean-up menu can be used to remove automatically generated data.

The **Delete current URI caches** will delete the uriLastModified.db file from the active certificates and CRLs folder. When a certificate, PKCS #7 file or CRL is downloaded from an HTTP server that uses the Last-Modified header, PITTv2 will store the URI and last modified value in a small SQLite database file named uriLastModified.db at the root of the certificates or CRLs folder (or a default certificates or CRLs folder if no folder has been specified). Values from this file are consulted prior to downloading an artifact. If the artifact has not changed, it is not downloaded. When the uriLastModified.db file contains values that correspond to files that have been deleted from the certificates or CRLs folder, certification path processing errors may occur. To avoid synchronization problems either delete the uriLastModified.db file or remove the entry using the **Uris** panel.

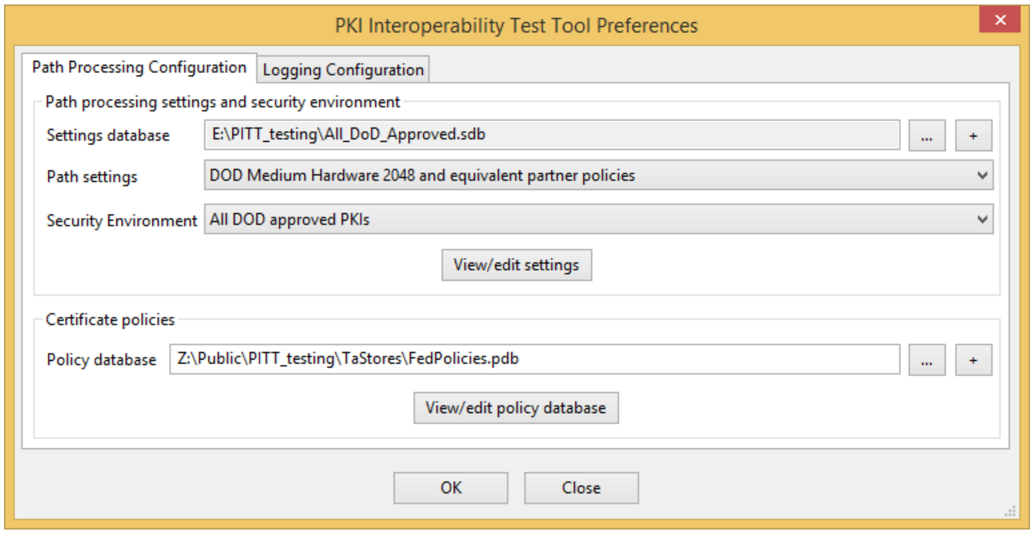
The **Delete temporary folder contents** option can be used to delete data written to the per-user temporary locations. Data may include certificates or CRLs downloaded when no certificates or CRLs folder has been specified or artifacts that could not be processed due to an error. Locations for each platform:

- Unix: ~/.PITTv2Data
- Windows: C:\Users\username\AppData\Local
- Mac: ~/ PITTv2

Options

Preferences

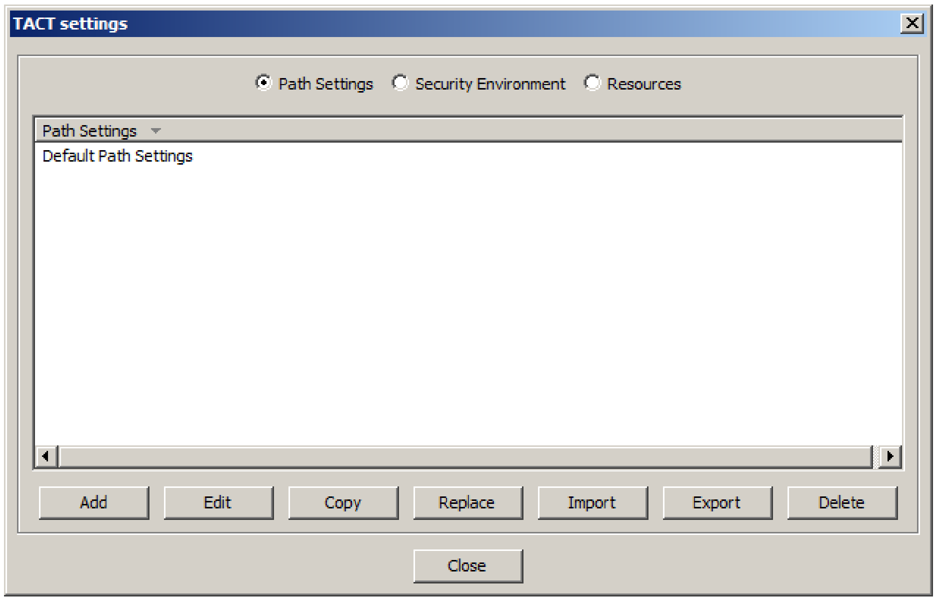
The **Preferences** menu item can be used to select a settings database, add/edit a new path settings configuration, add/edit a new security environment configuration, select/edit a policies database or configure logging.


PKI Interoperability Test Tool Preferences
Path Processing Configuration
Logging Configuration
Path processing settings and security environment
...
+
View/edit settings
Certificate policies
View/edit policy database
OK
Close

Settings database: E:\PITT_testing\All_DoD_Approved.sdb
Path settings: DOD Medium Hardware 2048 and equivalent partner policies
Security Environment: All DOD approved PKIs
Policy database: Z:\Public\PITT_testing\TaStores\FedPolicies.pdb

Figure 4 PITTv2 Preferences

TACT settings can be reviewed or edited by clicking the View/Edit button. A TACT Settings dialog will be displayed, as shown below. This dialog provides access to a collection of path settings definitions, security environment definitions and TACT resource definitions. The dialog features six buttons, each of which can be activated using a shortcut key pressed in tandem with the Alt key. The letter corresponding to each shortcut key appears underlined on each button when the Alt key is pressed.



[✓ Path Settings] Security Environment Resources

Path Settings
Default Path Settings

Figure 5 TACT settings editor

The radio buttons can be used to set which type of resource to review or edit. Path settings, security environment definitions and protected resource definitions can be edited using the TACT settings editor. Select the radio button corresponding to the type of item to view or configure. Each path settings item must have a unique name, as must each security environment. Each resource must have a unique path.

The **Add** button can be used to create a new item of the selected type in the TACT settings database. To edit an item, select the item to edit in the list box then click the **Edit** button. Path settings creation/editing and security environment creation/editing are covered in the Common Tasks section below.

Sometimes it is quicker to create a copy of an existing configuration item and make a small number of edits instead of creating a new item from scratch. To create a copy of an item, select the item to copy in the list box then click the **Copy** button and enter a new name for the item when prompted.

To import an item from a file, click the **Import** button then browse to the file to import.

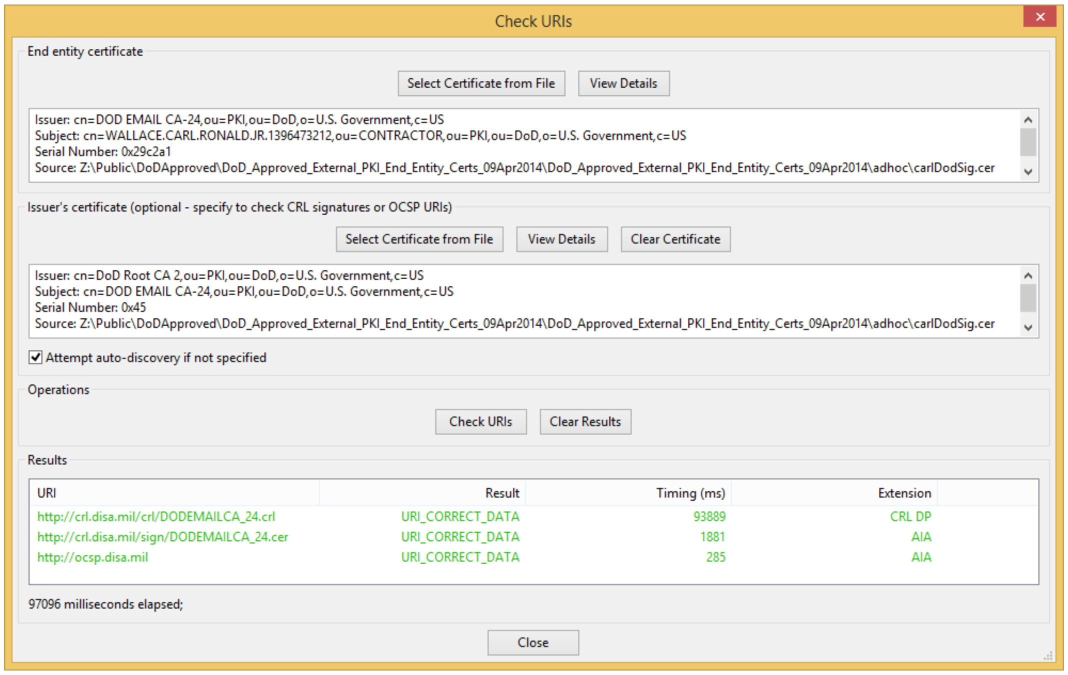
To export an item, select the item to export in the list box then click the **Export** button and navigate to the location where the file should be saved.

To delete an item, select the item to delete in the list box then click the **Delete** button.

Analysis

Check URIs in certificate

The **Check URIs in certificate** enables URIs contained in a certificate to be tested independent of certification path processing. Each HTTP and LDAP URI present in an authority information access (AIA), subject information access (SIA) or CRL distribution points (CRL DP) extension will be retrieved and evaluated relative to the certificate containing the extension. If the issuer's certificate is specified or **Attempt auto-discovery if not specified** is checked (and is successful) then CRL signatures will be verified[1] and OCSP AIA URIs will be tested. The following screen shot shows the Check URIs dialog with results following a check URI operation. Full certification path processing is not performed in support of CRL signature verification or OCSP processing. Use one of the panels to perform full path processing of revocation status providers in the context of a certificate validation.


Check URIs
End entity certificate
Select Certificate from File
View Details

Issuer: cn=DOD EMAIL CA-24,ou=PKI,ou=DoD,o=U.S. Government,c=US
Subject: cn=WALLACE.CARL.RONALD.JR.1396473212,ou=CONTRACTOR,ou=PKI,ou=DoD,o=U.S. Government,c=US
Serial Number: 0x29c2a1
Source: Z:\Public\DoDApproved\DoD_Approved_External_PKI_End_Entity_Certs_09Apr2014\DoD_Approved_External_PKI_End_Entity_Certs_09Apr2014\adhoc\carlDodSig.cer

Issuer's certificate (optional - specify to check CRL signatures or OCSP URIs)
Select Certificate from File
View Details
Clear Certificate

Issuer: cn=DoD Root CA 2,ou=PKI,ou=DoD,o=U.S. Government,c=US
Subject: cn=DOD EMAIL CA-24,ou=PKI,ou=DoD,o=U.S. Government,c=US
Serial Number: 0x45
Source: Z:\Public\DoDApproved\DoD_Approved_External_PKI_End_Entity_Certs_09Apr2014\DoD_Approved_External_PKI_End_Entity_Certs_09Apr2014\adhoc\carlDodSig.cer

[✓] Attempt auto-discovery if not specified

URI	Result	Timing (ms)	Extension
http://crl.disa.mil/crl/DODEMAILCA_24.crl	URI_CORRECT_DATA	93889	CRL DP
http://crl.disa.mil/sign/DODEMAILCA_24.cer	URI_CORRECT_DATA	1881	AIA
http://ocsp.disa.mil	URI_CORRECT_DATA	285	AIA

97096 milliseconds elapsed;

Figure 6 Check URIs dialog

To specify the certificate that contains the URIs to check, click the **Select Certificate from File** button, browse to a file containing a DER encoded certificate then click **Open**. The issuer name, serial number and subject name will be displayed in the **End entity**

[1] CRL signatures are only verified using the issuer's public key. To test scenarios involving CRLs signed with a new CA key or indirect CRLs, perform full certification path processing using one of the tabs.

certificate text box. To view the certificate using the Microsoft shell viewer, click the **View Details** button.

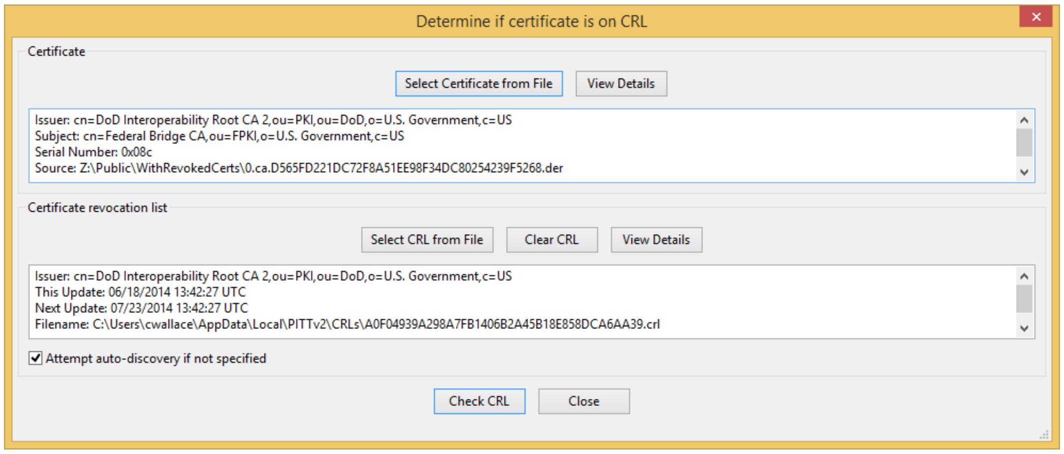
To check the URIs in the certificate, click the **Check URIs** button.

URI_NOT_AVAILABLE indicates that URI is not available.
URI_CORRECT_DATA indicates that URI points to correct information for the target certificate.
URI_INCORRECT_DATA indicates that URI points to incorrect information for the target certificate.
URI_WARNING indicates that URI points to a certificate collection that includes a self-signed certificate.

URI_UNKNOWN_ACCESS_METHOD indicates that SIA or AIA extension contains unknown access method.
URI_BLACKLISTED_HOST indicates the URI includes a host that is blacklisted in the default settings defined in Options->Preferences.
URI_CRL_NOT_CHECKED_NO_ISSUER_CERT indicates a CRL was obtained but was not inspected because the issuer's certificate was not available.

Determine if certificate is on CRL

The **Determine if certificate is on CRL** option can be used to determine if a given certificate is present on either a given CRL or a CRL retrieved using a CRL DP present in the certificate.


Determine if certificate is on CRL
Certificate
Select Certificate from File
View Details
Certificate revocation list
Select CRL from File
Clear CRL
View Details
Check CRL
Close

```
Issuer: cn=DoD Interoperability Root CA 2,ou=PKI,ou=DoD,o=U.S. Government,c=US
Subject: cn=Federal Bridge CA,ou=FPKI,o=U.S. Government,c=US
Serial Number: 0x08c
Source: Z:\Public\WithRevokedCerts\0.ca.D565FD221DC72F8A51EE98F34DC80254239F5268.der
```

```
Issuer: cn=DoD Interoperability Root CA 2,ou=PKI,ou=DoD,o=U.S. Government,c=US
This Update: 06/18/2014 13:42:27 UTC
Next Update: 07/23/2014 13:42:27 UTC
Filename: C:\Users\cwallace\AppData\Local\PITTv2\CRLs\A0F04939A298A7FB1406B2A45B18E858DCA6AA39.crl
```

☑ Attempt auto-discovery if not specified

Figure 7 Determine if certificate is on CRL dialog

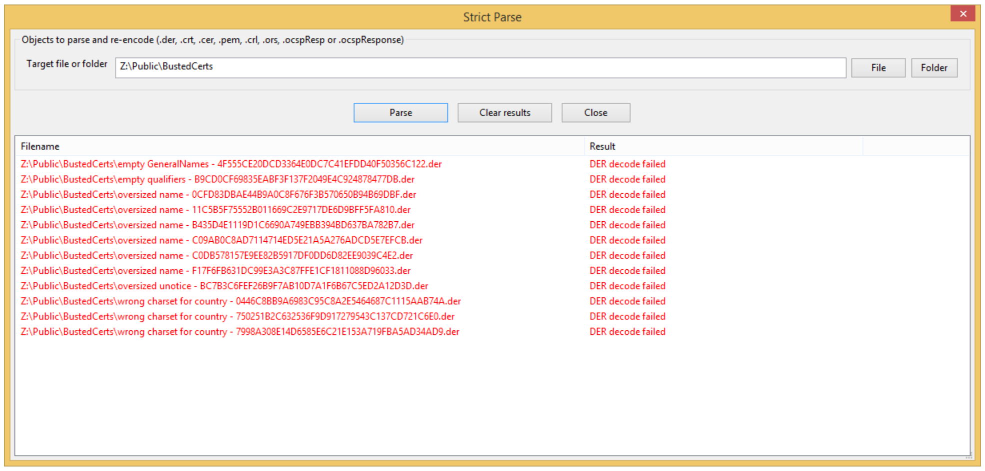
To specify the certificate to check, click the **Select Certificate from File** button, browse to a file containing a DER encoded certificate then click **Open**. The issuer name, serial number and subject name will be displayed in the **Certificate** box. To view the certificate using the Microsoft shell viewer, click the **View Details** button.

To specify the CRL to check, click the **Select CRL from File** button, browse to a file containing a DER encoded CRL then click **Open**. The issuer name, this update, next update and file name will appear in the **Certificate revocation list** box. Alternatively, check the **Attempt auto-discovery if not specified** check box to cause auto-retrieval of the CRL to be attempted.

Click the **Check CRL** button to inspect the specified CRL (or downloaded CRL) to see if the certificate appears on the CRL. Click the **Clear CRL** button to discard a previously used CRL before clicking **Check CRL** to ensure there are no compatibility issues when checking the certificate.

Check object encoding with strict parser

The parsing code used by PITTv2, CAPI and other products that use PKI include a variety of accommodations for handling artifacts that have encoding errors. For example, the root certificate programs (!) operated by Microsoft and Mozilla have accepted certificates with a variety of encoding errors including: usage of incorrect character sets for relative distinguished names, oversized fields, failure to honor size constraints on SEQUENCE OF or SET OF structures. These accommodations make it difficult to identify errors prior to generating artifacts of a new type. To help identify encoding errors, PITTv2 includes a strict ASN.1 parser that does not include accommodations for observed defects. The **Check object encoding with strict parser** option supports inspection of certificates, CRLs and OCSP responses.


Strict Parse
Objects to parse and re-encode (.der, .crt, .cer, .pem, .crl, .ors, .ocspResp or .ocspResponse)
File
Folder
Parse
Clear results
Close

Target file or folder: Z:\Public\BustedCerts

Filename	Result
Z:\Public\BustedCerts\empty GeneralNames - 4F555CE20DCD3364E0DC7C41EFDD40F50356C122.der	DER decode failed
Z:\Public\BustedCerts\empty qualifiers - B9CD0CF69835EABF3F137F2049E4C924878477DB.der	DER decode failed
Z:\Public\BustedCerts\oversized name - 0CFD83DBAE44B9A0C8F676F3B570650B94B69DBF.der	DER decode failed
Z:\Public\BustedCerts\oversized name - 11C5B5F75552B011669C2E9717DE6D9BFF5FA810.der	DER decode failed
Z:\Public\BustedCerts\oversized name - B435D4E1119D1C6690A749EBB394BD637BA782B7.der	DER decode failed
Z:\Public\BustedCerts\oversized name - C09AB0C8AD7114714ED5E21A5A276ADCD5E7EFCB.der	DER decode failed
Z:\Public\BustedCerts\oversized name - C0DB578157E9EE82B5917DF0DD6D82EE9039C4E2.der	DER decode failed
Z:\Public\BustedCerts\oversized name - F17F6FB631DC99E3A3C87FFE1CF1811088D96033.der	DER decode failed
Z:\Public\BustedCerts\oversized unotice - BC7B3C6FEF26B9F7AB10D7A1F6B67C5ED2A12D3D.der	DER decode failed
Z:\Public\BustedCerts\wrong charset for country - 0446C8BB9A6983C95C8A2E5464687C1115AAB74A.der	DER decode failed
Z:\Public\BustedCerts\wrong charset for country - 750251B2C632536F9D917279543C137CD721C6E0.der	DER decode failed
Z:\Public\BustedCerts\wrong charset for country - 7998A308E14D6585E6C21E153A719FBA5AD34AD9.der	DER decode failed

Figure 8 Strict Parse dialog

To use the dialog, click the **File** or **Folder** button and browse to a certificate file or folder then click the **Parse** button. Each supported file type that is found will be inspected and an indication of success or failure will be written to the list box. Detailed information is not available via the dialog, but the PITTv2 logs typically have additional information regarding the nature of the defect. Two basic failure modes are possible: failure to parse, failure to re-encode using DER to produce the same encoding. The logs will have information on decoding failures but do not identify the fields that are not DER encoded when re-encoding fails to produce the same encoding. Manual analysis is required in such cases. Typically, these errors are due to BIT STRING fields that have spurious leading bits that are disallowed by DER.

List CAPI revocation status providers

The **List CAPI revocation status providers** option is a shortcut to reading the registry to determine which revocation status providers are present on a system. This option is only enabled on Windows systems. The list presented is read from:

HKEY_LOCAL_MACHINE\Software\Microsoft\Cryptography\OID\EncodingType 1\CertDllVerifyRevocation\DEFAULT

Help

About

The **About** option displays information about PITTv2, including version information.

Panels

On Windows platforms, PITTv2 provides three panels that can be used for different purposes: All Paths, CAPI and Reports. On non-Windows platforms, PITTv2 provides two panels: All Paths and Reports. Each of these panels is described in the sections below.

The **All Paths** and **Reports** panels both use the settings database specified in the **Options->Preferences** menu item. Path settings and security environments can be edited via the **Edit** buttons on the panels, but additions and deletions must be performed via **Options->Preferences**.

All Paths

The **All Paths** panel can be used to build and validate paths to one or more certificates. When a certificate file is specified in the **Target certificate file or folder** box and **Return first path only**, the results list will include as many certification paths as can be found for the certificate when **Validate** is clicked. When a folder is specified, the results list will include as many certification paths as can be found for the .cer, .crt, .der or .pem files found while recursively processing the folder. When **Return first path only** is checked, only the first path found for each certificate that is processed will be added to the results list.

When **Check URIs during path processing** is checked, each URI included in a target certificate is checked similar to the checks performed via the **Analysis->Check URIs in certificate** menu item.

Typically, PITT fails when an error is encountered during path validation. This can make it difficult to identify all problems during testing. When **Display all path validation errors** is checked, PITT will continue past the first error and display all the path validation errors encountered during path processing in the path log. This will allow users to see all the possible issues with the candidate path in cases there is more than one.

PITT normally catches revocation status information to minimize overhead. This can make it difficult to identify source of revocation information in some of the paths. When **Clear revocation status cache for each path** is checked, revocation cache will be cleared for each path. This will allow users to see the nature of revocation information for each certificate.

PITTv1 did a full path validation on OCSP responder certificates, because the specification doesn’t require this PITTv2 does not do full path validation on OCSP responder certificates. During testing this can be helpful to flag down issues with OCSP responders. When **Validate OCSP responder certificates** is checked, all the OCSP responder certificates encountered during path validation will be validated and validation results added to the path log. This setting has to be used in conjunction with **Clear revocation status cache for each path** to make OCSP responder certificates available for validation.

The **Validate** button initiates certification path processing operations. As certification paths are found and validated, results are written to the list. To stop processing prior to completion, click the **Cancel** button and processing will terminate at the next opportunity. To inspect a specific result, double click the row to display a textual log. The resulting dialog displays information about the certification path and features buttons that allow saving artifacts from the path. Alternatively, right click a row and choose **View Results**, **Save Artifacts** or **List AIAs** from the resulting context menu.

The **Clear Results** button clears the contents in the results list.

The **Clear rev. status cache** resets the revocation status cache maintained by PITTv2, causing subsequent revocation status operations to utilize and available revocation status information source (i.e., local CRLs, OCSP or remote CRLs). Revocation status values of good or revoked are automatically cached for 10 minutes, avoiding the need to repeatedly query an OCSP responder, for example, while building and validating all certification paths for a given target.

The **Save map data** button saves the PKI graph currently loaded by the panel to a .dot file for review using Graphviz.

The **List AIAs** button displays a list of URIs from the AIA extensions found in the certificates associated with the results displayed in the results list. When the results list is empty, no AIA list is displayed.

The **Save Path Logs from Results** button allows a single text file with a concatenation of the logs associated with the results displayed in the results list.

The **Save Artifacts from Results** button allows artifacts associated with the results displayed in the results list to be saved in one action. Artifacts are written to a selected folder within subfolders named using the row ID.

The **Check for duplicates** button is a diagnostic tool to confirm no duplicate paths were returned during certification path processing.

CAPI

The **CAPI** panel is similar to the **All Paths** panel except it uses Microsoft CAPI functions to perform certification path processing, using certificates, trust anchors, revocation status determination mechanisms, etc. available to the current user.

The **Build and Validate All Paths** button is similar to the **Validate** button on the **All Paths** panel. Unlike **All Paths**, there is no option to limit results on first path only.

The **Clear Results** button clears the contents in the results list.

The **Save Path Logs from Results** button allows a single text file with a concatenation of the logs associated with the results displayed in the results list.

Reports

The **Reports** panel allows generation of various reports relative to a given path settings configuration and security environment configuration. Reports supported in this release include:

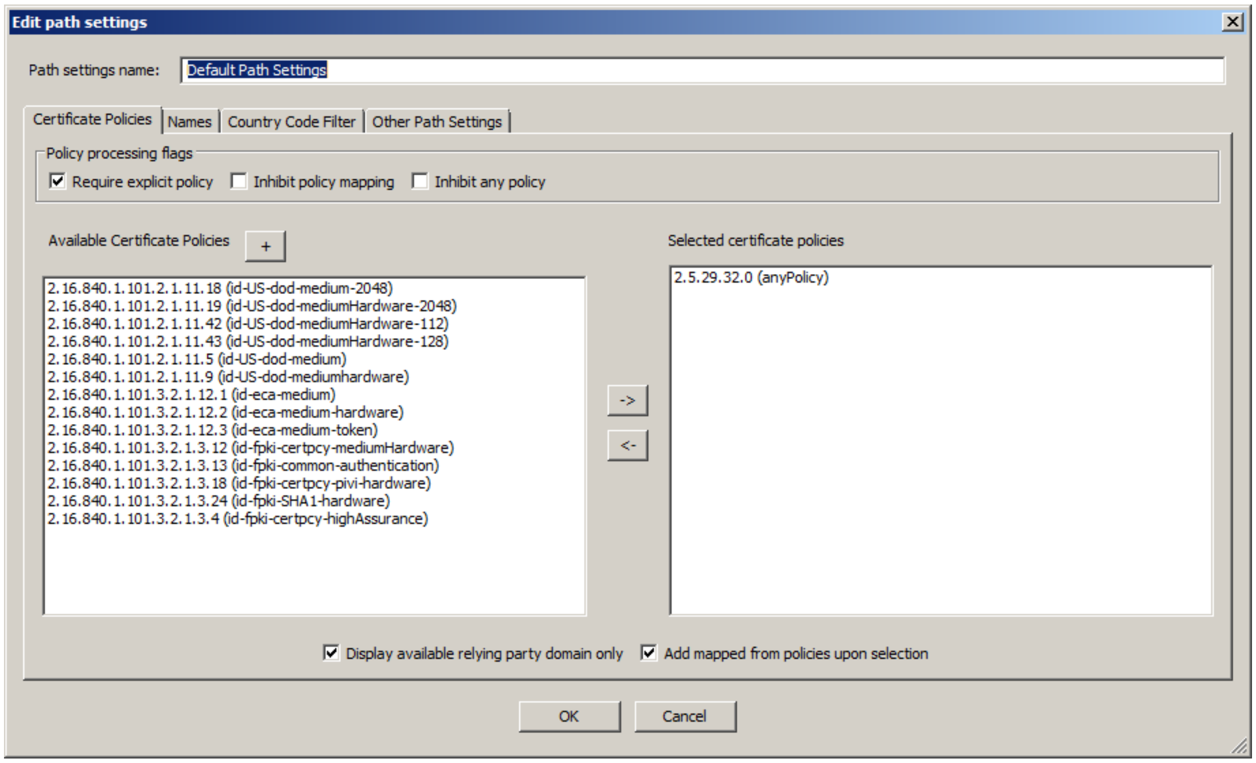
- Connected trust anchors
- Disconnected trust anchors
- Connected certification authorities
- Disconnected certification authorities

When the **Reports** panel is loaded or when the selected path settings or security environment setting is changed, a new PKI graph is constructed. The reports are based on the current graph. Connected trust anchors are trust anchors present in the graph with at least one child CA. Disconnected trust anchors are trust anchors with no child CAs. Connected certification authorities are CAs with at least one parent or child. Disconnected certification authorities are CAs with no parent and no children.

Common Tasks

Edit path settings

Path settings can be viewed and edited in several contexts in the TACT management applications. When editing a TACT settings database, the path settings editor includes a **Path settings name** field that can be used to change the name of a path settings configuration item. In other contexts, the name cannot be changed and the **Path settings name** field is absent. The figures in this section all include the **Path settings name** field, as shown below.


Edit path settings
Certificate Policies | Names | Country Code Filter | Other Path Settings
+
->
<-
OK
Cancel

Path settings name: Default Path Settings

Policy processing flags

[✓] Require explicit policy [ ] Inhibit policy mapping [ ] Inhibit any policy

Available Certificate Policies

2.16.840.1.101.2.1.11.18 (id-US-dod-medium-2048)
2.16.840.1.101.2.1.11.19 (id-US-dod-mediumHardware-2048)
2.16.840.1.101.2.1.11.42 (id-US-dod-mediumHardware-112)
2.16.840.1.101.2.1.11.43 (id-US-dod-mediumHardware-128)
2.16.840.1.101.2.1.11.5 (id-US-dod-medium)
2.16.840.1.101.2.1.11.9 (id-US-dod-mediumhardware)
2.16.840.1.101.3.2.1.12.1 (id-eca-medium)
2.16.840.1.101.3.2.1.12.2 (id-eca-medium-hardware)
2.16.840.1.101.3.2.1.12.3 (id-eca-medium-token)
2.16.840.1.101.3.2.1.3.12 (id-fpki-certpcy-mediumHardware)
2.16.840.1.101.3.2.1.3.13 (id-fpki-common-authentication)
2.16.840.1.101.3.2.1.3.18 (id-fpki-certpcy-pivi-hardware)
2.16.840.1.101.3.2.1.3.24 (id-fpki-SHA1-hardware)
2.16.840.1.101.3.2.1.3.4 (id-fpki-certpcy-highAssurance)

Selected certificate policies

2.5.29.32.0 (anyPolicy)

[✓] Display available relying party domain only [✓] Add mapped from policies upon selection

Figure 9 Path settings editor

In addition to the **Path settings name** field, the path settings editor features three tabs: **Certificate Policies**, **Names**, **Other Path Settings**. The contents of each tab are described below.

Certificate Policies

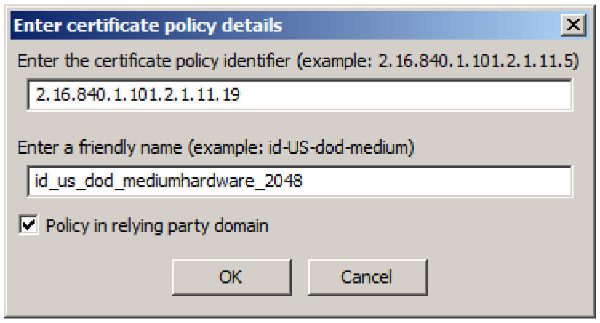
The **Certificate Policies** tab is used to define four certificate policy-related RFC 5280 certification path validation inputs:

- The **Selected certificate policies** list box contents are used as the user-initial-policy-set input.

- The **Require explicit policy** checkbox is used as the initial-explicit-policy input.
- The **Inhibit policy mapping** checkbox is used as the initial-policy-mapping-inhibit input.
- The **Inhibit any policy** is used as the initial-any-policy-inhibit input.

The **Available Certificate Policies** list is populated with the contents of a TACT certificate policies database. To add a policy to the **Selected certificate policies** list select the policy to add in the **Available Certificate Policies** list box and click the **->** button. To remove a policy from the **Selected certificate policies** list select the policy to remove in the **Selected Certificate Policies** list box and click the **<-** button. To add a policy to the **Available Certificate Policies** list box, click the **+** button and enter the desired information in the resulting dialog.

The **Display available relying party domain only** check box can be used to limit the number of policies displayed in the **Available Certificate Policies** list box. For this to be useful, the TACT certificate policies database must be defined with appropriate certificate policies marked as being in the relying party domain. The policies so marked will vary from one enterprise to another. For example, DOD certificate policies will be marked as relying party domain for usage by a DOD application but not marked as being in the relying party domain for usage by an application operated by a partner enterprise. The figure below shows the policy definition dialog.


Enter certificate policy details

Enter the certificate policy identifier (example: 2.16.840.1.101.2.1.11.5)

2.16.840.1.101.2.1.11.19

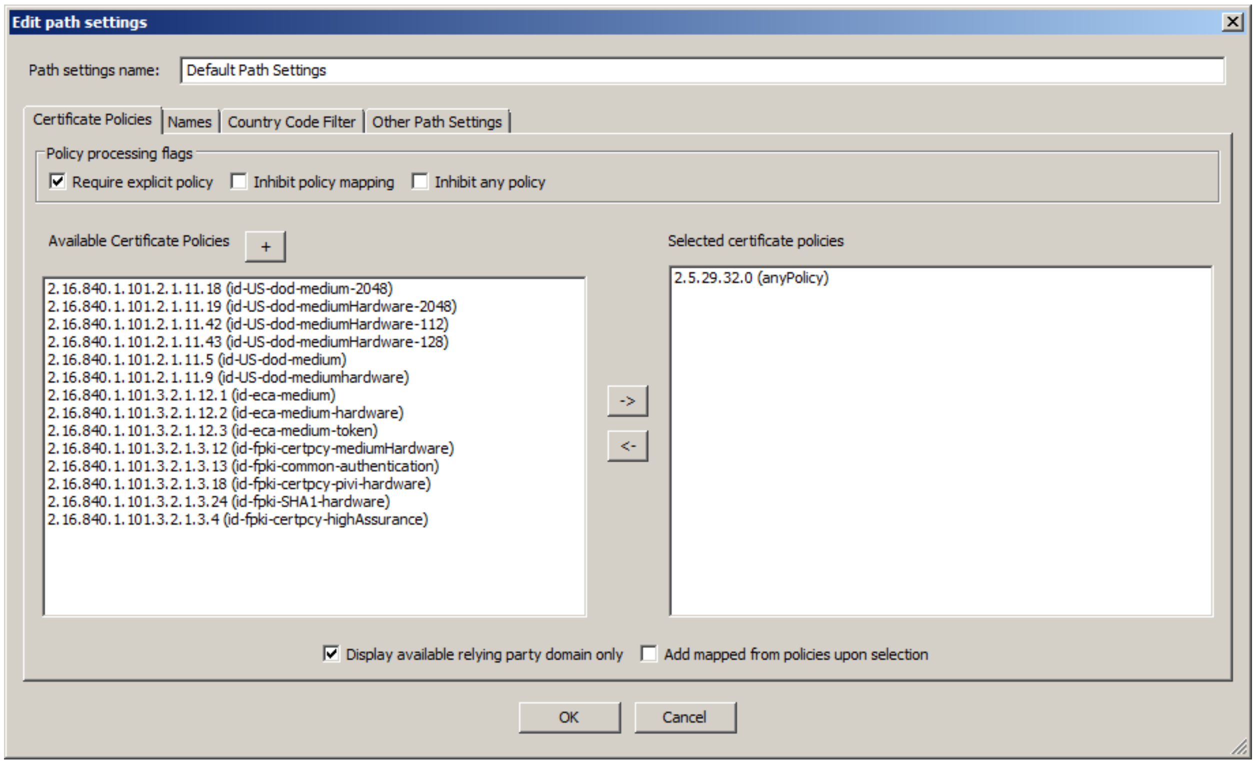
Enter a friendly name (example: id-US-dod-medium)

id_us_dod_mediumhardware_2048

[✓] Policy in relying party domain

Figure 10 Certificate policy definition

The figure below shows the **Certificate Policies** tab with the **Display available relying party domain only** check box checked while using a TACT certificate policies database that has a set of DOD policies marked as being in the relying party domain.


Edit path settings

Path settings name: Default Path Settings

Certificate Policies | Names | Country Code Filter | Other Path Settings

Policy processing flags

[✓] Require explicit policy [ ] Inhibit policy mapping [ ] Inhibit any policy

Available Certificate Policies +

```
2.16.840.1.101.2.1.11.18 (id-US-dod-medium-2048)
2.16.840.1.101.2.1.11.19 (id-US-dod-mediumHardware-2048)
2.16.840.1.101.2.1.11.42 (id-US-dod-mediumHardware-112)
2.16.840.1.101.2.1.11.43 (id-US-dod-mediumHardware-128)
2.16.840.1.101.2.1.11.5 (id-US-dod-medium)
2.16.840.1.101.2.1.11.9 (id-US-dod-mediumhardware)
2.16.840.1.101.3.2.1.12.1 (id-eca-medium)
2.16.840.1.101.3.2.1.12.2 (id-eca-medium-hardware)
2.16.840.1.101.3.2.1.12.3 (id-eca-medium-token)
2.16.840.1.101.3.2.1.3.12 (id-fpki-certpcy-mediumHardware)
2.16.840.1.101.3.2.1.3.13 (id-fpki-common-authentication)
2.16.840.1.101.3.2.1.3.18 (id-fpki-certpcy-pivi-hardware)
2.16.840.1.101.3.2.1.3.24 (id-fpki-SHA1-hardware)
2.16.840.1.101.3.2.1.3.4 (id-fpki-certpcy-highAssurance)
```

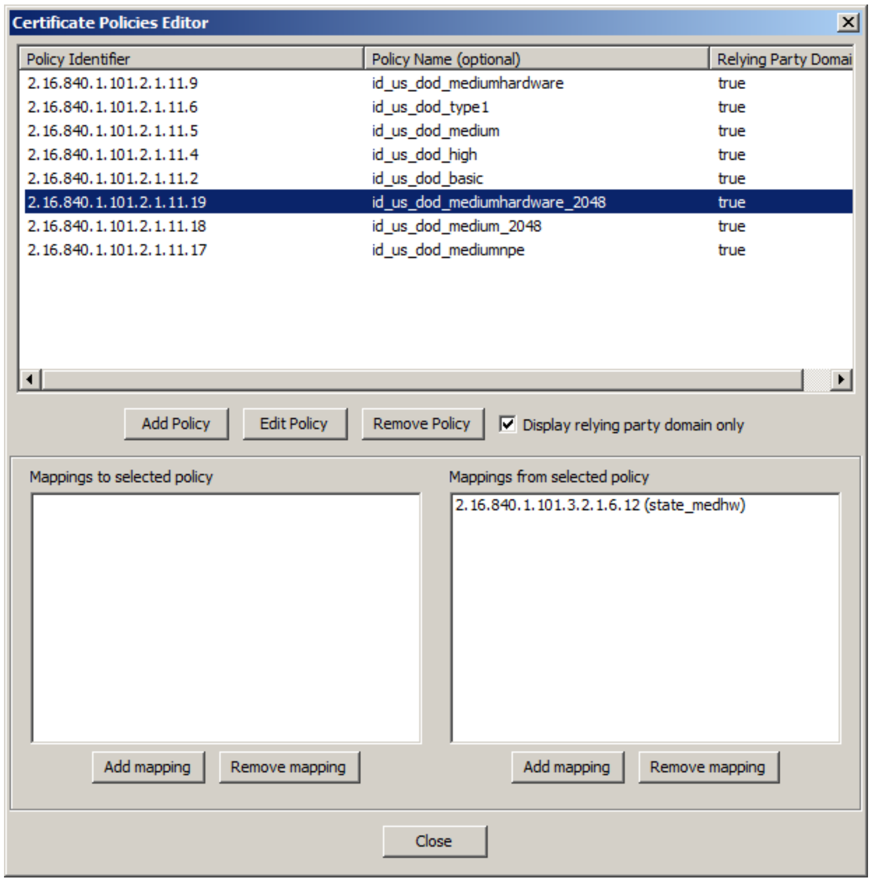
Selected certificate policies

```
2.5.29.32.0 (anyPolicy)
```

[✓] Display available relying party domain only [ ] Add mapped from policies upon selection

Figure 11 Relying party policies only displayed

The **Add mapped from policies upon selection** can be used to move a set of policies from the **Available Certificate Policies** list to the **Selected Certificate Policies** list. When this option is checked, any certificate policies that have been mapped to a policy selected in the **Available Certificate Policies** list will be moved to the **Selected Certificate Policies** list when the **->** button is clicked. The figure below shows the Certificate Policy Editor dialog with a DOD policy selected that has one associated mapping.



Policy Identifier	Policy Name (optional)	Relying Party Domai
2.16.840.1.101.2.1.11.9	id_us_dod_mediumhardware	true
2.16.840.1.101.2.1.11.6	id_us_dod_type1	true
2.16.840.1.101.2.1.11.5	id_us_dod_medium	true
2.16.840.1.101.2.1.11.4	id_us_dod_high	true
2.16.840.1.101.2.1.11.2	id_us_dod_basic	true
2.16.840.1.101.2.1.11.19	id_us_dod_mediumhardware_2048	true
2.16.840.1.101.2.1.11.18	id_us_dod_medium_2048	true
2.16.840.1.101.2.1.11.17	id_us_dod_mediumnpe	true

☑ Display relying party domain only

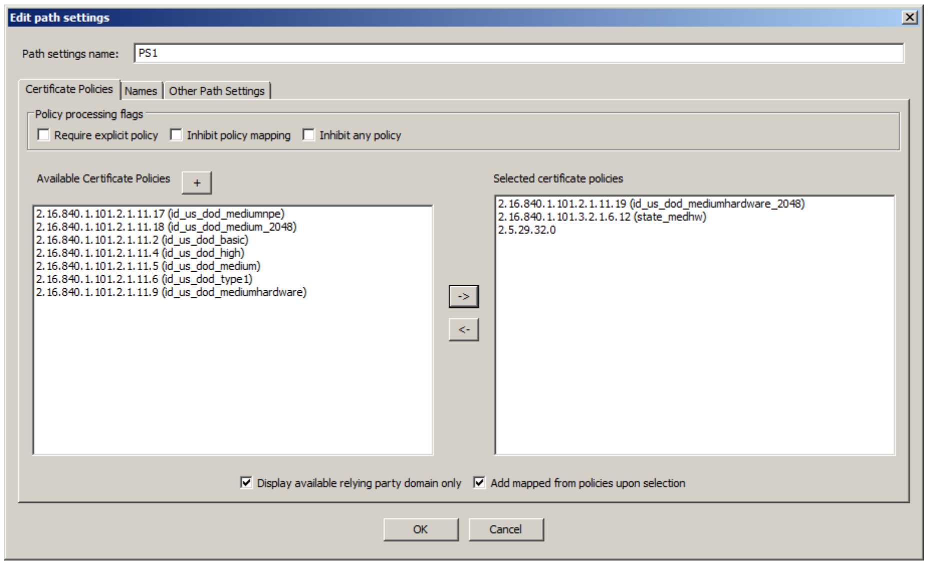
Mappings to selected policy

Mappings from selected policy

2.16.840.1.101.3.2.1.6.12 (state_medhw)

Figure 12 Example policy definition with associated mapping

The figure below shows the result of moving this policy when the **Add mapped to policies upon selection** option is checked. Note, the **Selected certificate policies** list always shows a complete list of policies that have been selected. Only the display of the **Available certificate policies** list can be limited using the **Display available relying party domain only** check box.



Path settings name: PS1

Policy processing flags

- [ ] Require explicit policy
- [ ] Inhibit policy mapping
- [ ] Inhibit any policy

Available Certificate Policies

```
2.16.840.1.101.2.1.11.17 (id_us_dod_mediumnpe)
2.16.840.1.101.2.1.11.18 (id_us_dod_medium_2048)
2.16.840.1.101.2.1.11.2 (id_us_dod_basic)
2.16.840.1.101.2.1.11.4 (id_us_dod_high)
2.16.840.1.101.2.1.11.5 (id_us_dod_medium)
2.16.840.1.101.2.1.11.6 (id_us_dod_type1)
2.16.840.1.101.2.1.11.9 (id_us_dod_mediumhardware)
```

Selected certificate policies

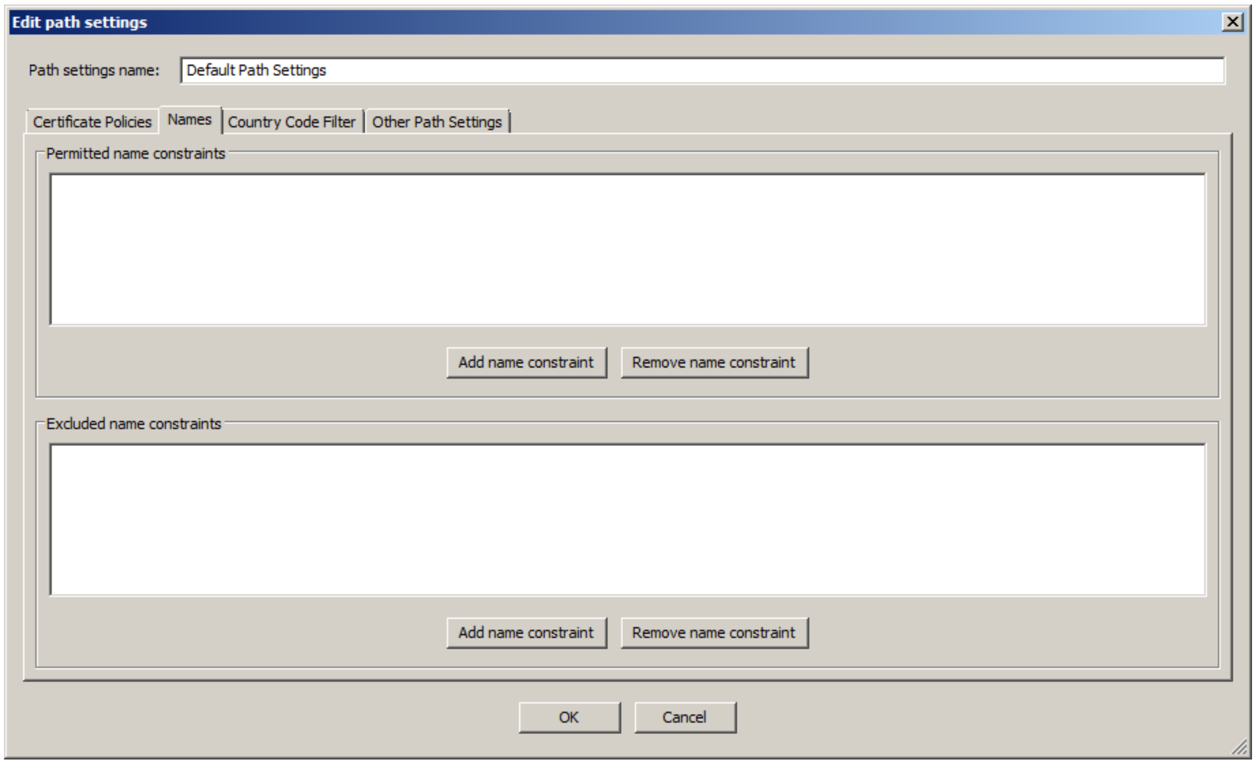
```
2.16.840.1.101.2.1.11.19 (id_us_dod_mediumhardware_2048)
2.16.840.1.101.3.2.1.6.12 (state_medhw)
2.5.29.32.0
```

- [x] Display available relying party domain only
- [x] Add mapped from policies upon selection

Figure 13 Auto-selection of mapped policies

Names

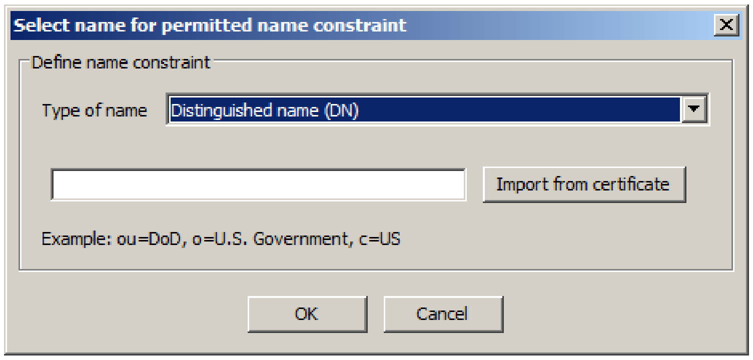
Permitted and excluded name constraints can be defined on the **Names** tab, shown below.



Path settings name: Default Path Settings

Figure 14 Names tab

To add a permitted name, click the **Add name constraint** button in the **Permitted name constraints** box. A dialog like the one shown below will be displayed. This dialog allows specification of the following name forms: Distinguished names, URIs, DNS names, RFC 822 names and UPNs.


Select name for permitted name constraint

Define name constraint

Type of name: Distinguished name (DN)

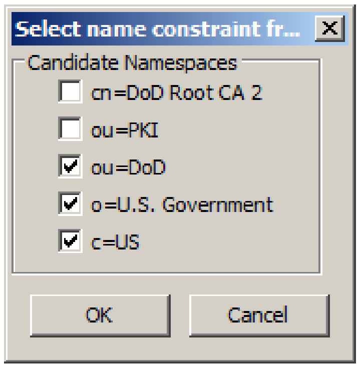
Import from certificate

Example: ou=DoD, o=U.S. Government, c=US

OK Cancel

Figure 15 Name constraint entry dialog

To import a distinguished name, click the **Import from certificate** button, browse to a certificate and click on the desired Relative Distinguished Name (RDN). The dialog below shows selections of the following namespace as imported from the DOD Root CA2 certificate: ou=DoD, o=U.S. Government, c=US.


Select name constraint fr...

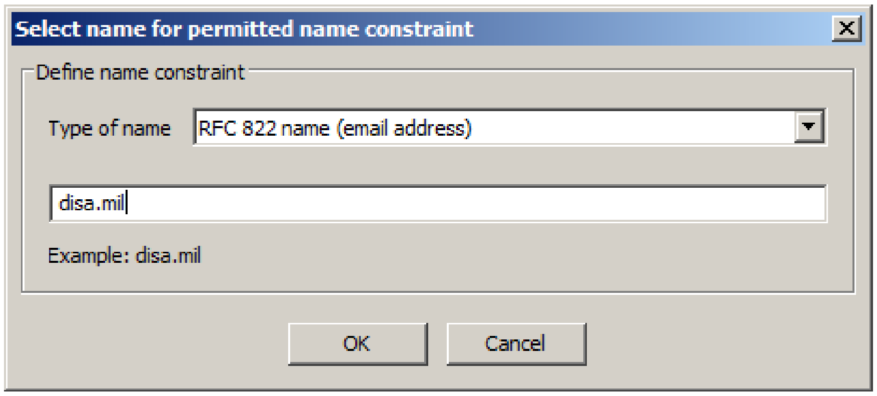
Candidate Namespaces

- [ ] cn=DoD Root CA 2
- [ ] ou=PKI
- [✓] ou=DoD
- [✓] o=U.S. Government
- [✓] c=US

OK Cancel

Figure 16 Importing a distinguished name

All other name forms are simply entered into the text box. The figure below shows specification of an RFC 822 name form.



Type of name: RFC 822 name (email address)

disa.mil

Example: disa.mil

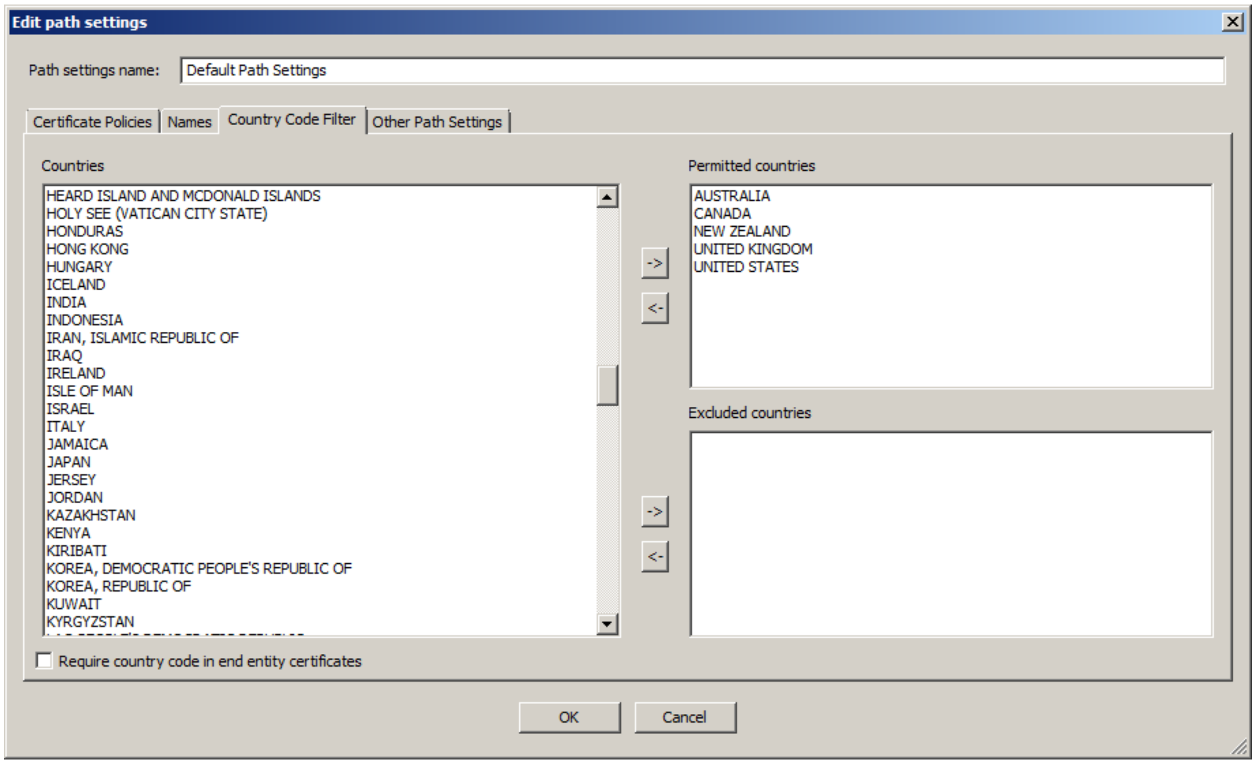
Figure 17 Specifying an RFC 822 name constraint

Use the **Type of name** drop list to select alternative name forms. To remove a previously specified name, select the name to remove then click the **Remove name constraint** button.

When defining name constraints, it is important to keep in mind that the constraints will be processed per RFC 5280 and RFC 5937 rules. This can result in some relatively counterintuitive conditions. For example, a specific individual can be denied access using a single excluded name constraint but a specific individual cannot be granted accessing using a single permitted name constraint. This is due to the nature of path processing which would result in an intermediate CA certificate failing to satisfy the permitted name constraint. Similarly, name constraints only apply to certificates that include a given name form. Thus, it is not possible to permit, for example, all mil UPNs because this would also allow certificates that do not contain a UPN.

Country Code Filter

The **Country Code Filter** tab, shown below, features three list boxes that allow for specification of sets of permitted or excluded country codes and a check box that can be used to require the presence of a country code value in end entity certificates in order for the certificate to be successfully validated.



Path settings name: Default Path Settings

Countries

HEARD ISLAND AND MCDONALD ISLANDS
HOLY SEE (VATICAN CITY STATE)
HONDURAS
HONG KONG
HUNGARY
ICELAND
INDIA
INDONESIA
IRAN, ISLAMIC REPUBLIC OF
IRAQ
IRELAND
ISLE OF MAN
ISRAEL
ITALY
JAMAICA
JAPAN
JERSEY
JORDAN
KAZAKHSTAN
KENYA
KIRIBATI
KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF
KOREA, REPUBLIC OF
KUWAIT
KYRGYZSTAN

Permitted countries

AUSTRALIA
CANADA
NEW ZEALAND
UNITED KINGDOM
UNITED STATES

Excluded countries

☐ Require country code in end entity certificates

Figure 18 Country code filter dialog

The **Countries** list box features a list of countries from ISO 3166: http://www.iso.org/iso/home/standards/country_codes/country_names_and_code_elements_txt.htm. One or more countries can be selected to move into either the Permitted Countries or Excluded Countries list boxes using the appropriate **<-** or **->** buttons. The **Require country code in end entity certificates** check box can be used to require all end entity certificates to include a country code in order to be successfully validated.

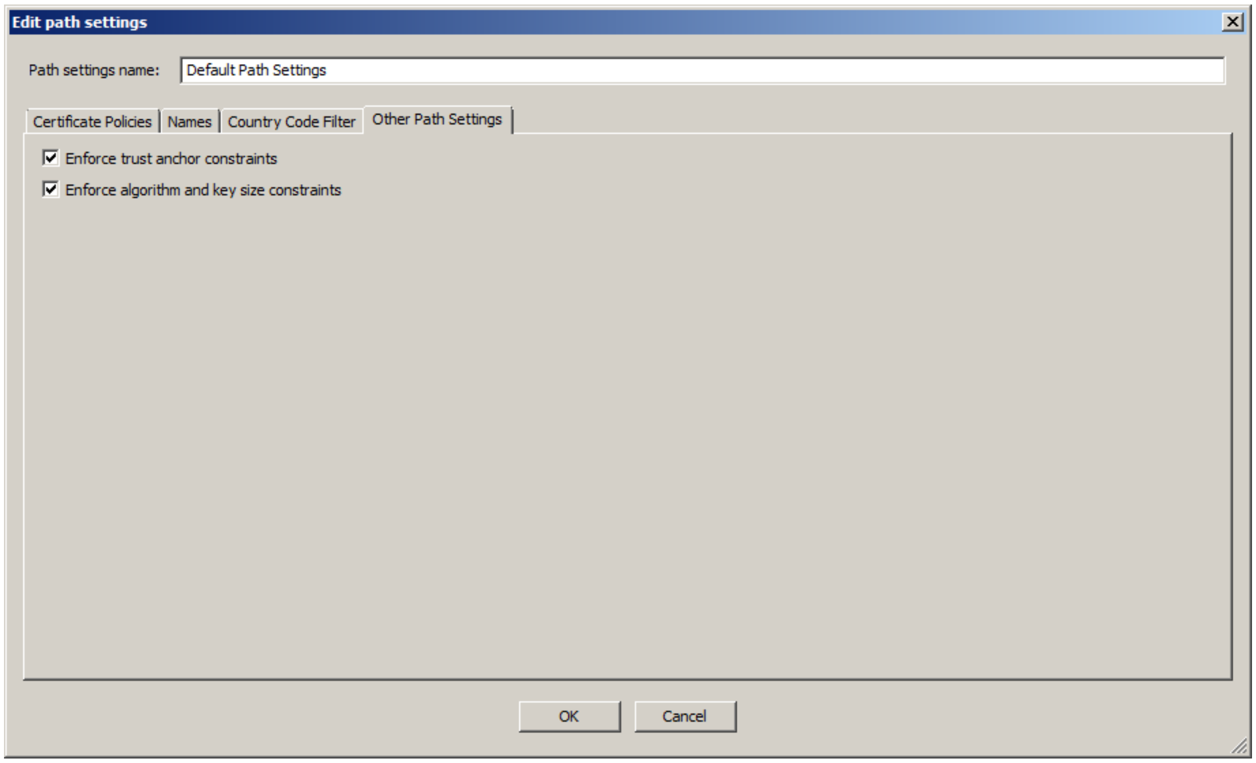
During certification path validation, when the **Require country code in end entity certificates** box is not checked, end entity certificates that do not contain a country code value in the subjectDirectoryAttributes extension pass the country code filter check. When the **Require country code in end entity certificates** box is checked, end entity certificates that do not contain a country code value in the subjectDirectoryAttributes extension fail the country code filter check.

When the **Excluded countries** list is not empty, the contents of the country code value in the subjectDirectoryAttributes extension in end entity certificates are evaluated. Certificates containing a value that matches an item in the **Excluded countries** list fail the country code filter check.

When the **Permitted countries** list is not empty, the contents of the country code value in the subjectDirectoryAttributes extension in end entity certificates are evaluated. Certificates that do not contain at least one value that matches an item in the **Permitted countries** list fail the country code filter check.

Other Path Settings

The **Other Path Settings** tab, shown below, features two check boxes that allow processing of information from an associated Security Environment to be optionally enabled or disabled.


Edit path settings

Path settings name: Default Path Settings

Certificate Policies | Names | Country Code Filter | Other Path Settings

☑ Enforce trust anchor constraints

☑ Enforce algorithm and key size constraints

Figure 19 Other Path Settings panel

The **Enforce trust anchor constraints** check box corresponds to the enforceTrustAnchorConstraints certificate path validation input defined in RFC 5937. When this box is checked, trust anchor-based constraints are enforced per RFC 5937. When it is not checked, trust anchor-based constraints are not enforced. The **Enforce algorithm and key size constraints** check box enables enforcement of cryptographic algorithm and key size constraints defined in the associated security environment. When this box is checked, algorithm and key size constraints are enforced. When this box is not checked, algorithm and key size constraints are not enforced.

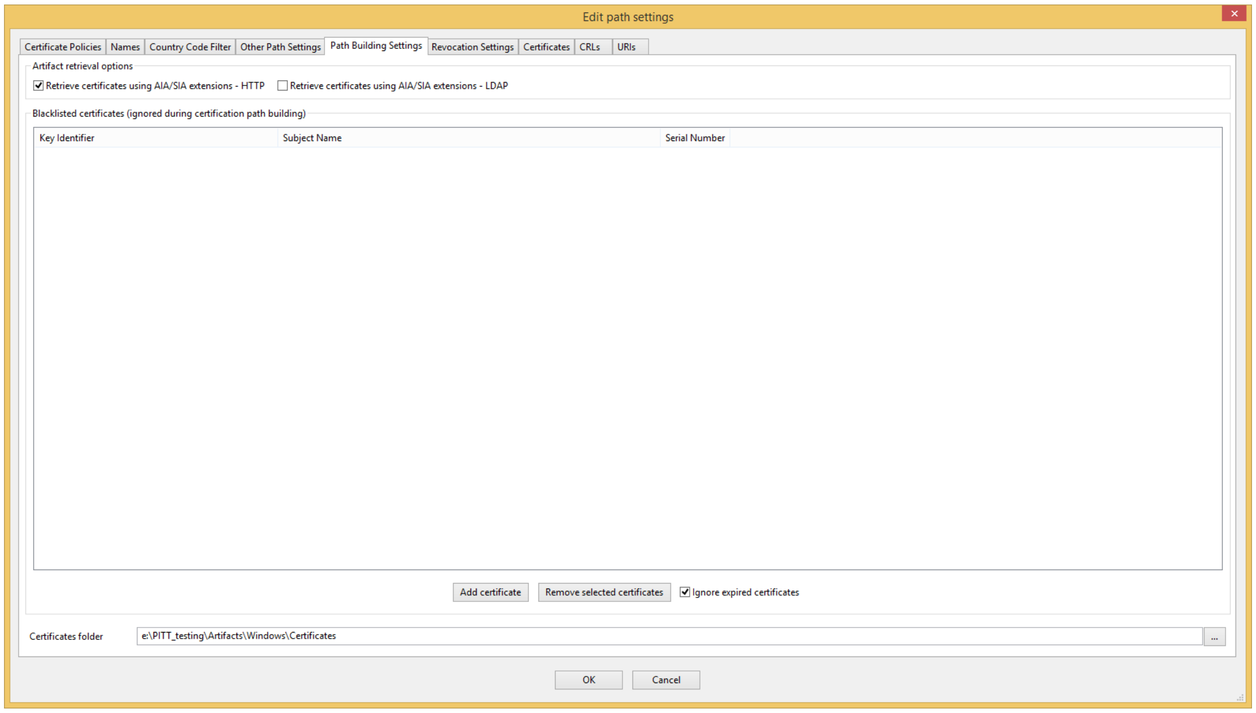
Dynamic path building and revocation status determination settings

These settings apply only to TACT v1.2 and later and only to TACT as integrated with mod_nss. The interfaces to edit the settings are present on all platforms, but the settings are only exercised within TACT when integrated with mod_nss[2].

Path Building Settings

The **Path Building Settings** tab defines the basic rules for building certificate graphs and obtaining certificates from remote sources. TACT can obtain certificates from a local file folder, from remote servers accessed via HTTP or from remote servers accessed via LDAP. The **Retrieve certificates using AIA/SIA extensions – HTTP** and **Retrieve certificates using AIA/SIA extensions – LDAP** settings determine if remote sources are used. If both are unchecked, no remote sources are consulted to obtain additional certificates. The **Certificates folder** setting identifies the local file folder containing certificates to use when preparing a certificate graph for use during certification path discovery. The folder will be recursively searched for .der, .cer, .crt and .pem files. The **Ignore expired certificates** setting determines whether expired certificates are included in a graph used for certification path discovery. When this option is checked, expired CA certificates are not used. The **Blacklisted certificates** box identifies any additional certificates that should be excluded from consideration during certification path discovery. Click the **Add certificate** button to browse to a certificate file to add to this list or select a certificate in the list and click the **Remove selected certificates** button to remove a certificate from the list.

[2] The settings may also be used by other utilities, like the PKI Interoperability Test Tool version 2 (PITTv2).



Artifact retrieval options

[✓] Retrieve certificates using AIA/SIA extensions - HTTP [ ] Retrieve certificates using AIA/SIA extensions - LDAP

Blacklisted certificates (ignored during certification path building)

Key Identifier	Subject Name	Serial Number

[✓] Ignore expired certificates

Certificates folder: e:\PITT_testing\Artifacts\Windows\Certificates

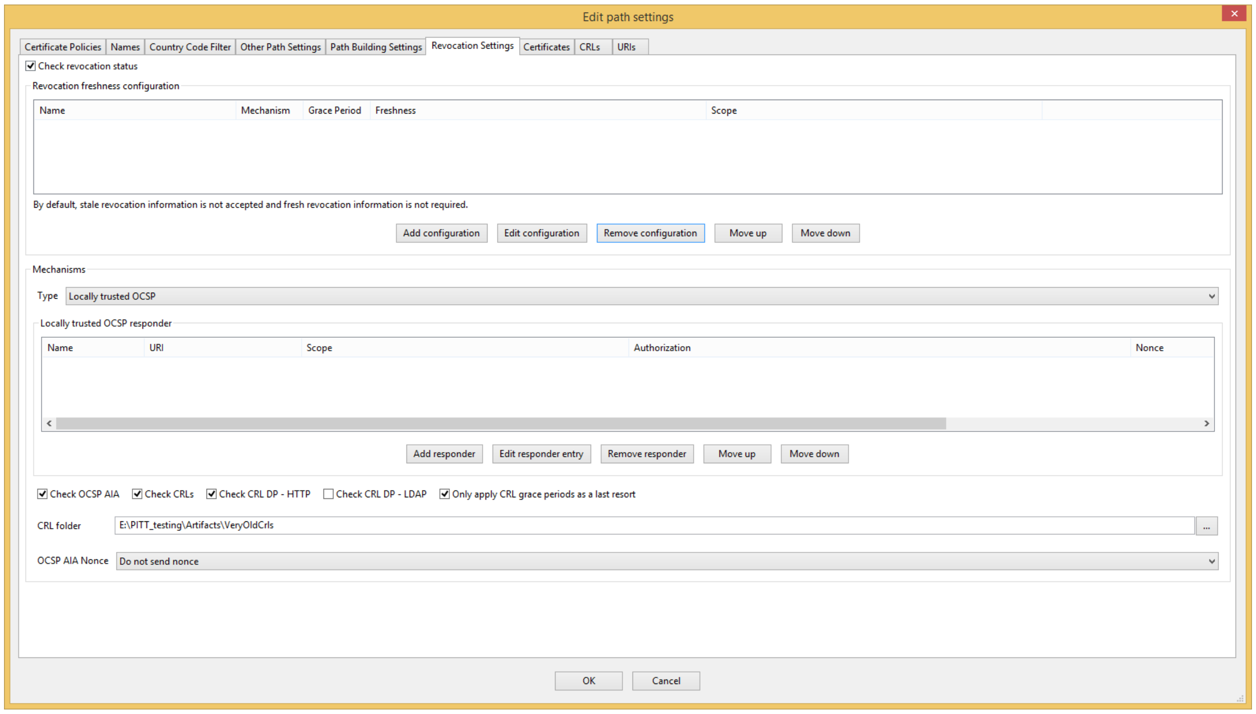
Figure 20 Path building settings

Revocation Settings

The **Revocation Settings** tab is used to configure the mechanisms used to determine the revocation status of certificates and how CRL time values are processed. The **Check revocation status** setting determines whether revocation status determination checks will be performed at all. When this is checked, revocation status will be checked in accord with other settings. When this is not checked, no attempt will be made to determine the revocation status of any certificate. The check boxes below the **Mechanisms** list control the commonly used options.

When the **Check OCSP AIA** setting is checked, OCSP responders identified in the authorityInformationAccess (AIA) extension of a certificate being validated may be consulted for revocation status information. When processing an OCSP response obtained via a URI from an AIA extension the response must be verified using the CA certificate used to verify the certificate containing the AIA extension or the responder's certificate must be verified using the CA certificate used to verify the certificate containing the AIA extension. Where the responder is not the CA, the responder's certificate must be included in the response. The **OCSP AIA Nonce** setting determines how nonces are used (or not used) in requests sent to responder identified in an AIA extension.

When the **Check CRLs** setting is checked, CRLs may be processed to determine revocation status information. If a **CRL folder** is specified, locally available CRLs will always be consulted before OCSP or remote CRL locations. Locations identified in a CRL distribution points (CRL DPs) extension will be consulted if the **Check CRL DP – HTTP** or **Check CRL DP – LDAP** settings are checked, with HTTP sources being consulted first if both options are checked. If a revocation freshness configuration is defined to allow stale CRLs to be used and the **Only apply CRL grace periods as a last resort** is checked, then an attempt to retrieve fresh information will be made before relying on stale information.



☑ Check revocation status

By default, stale revocation information is not accepted and fresh revocation information is not required.

Type: Locally trusted OCSP

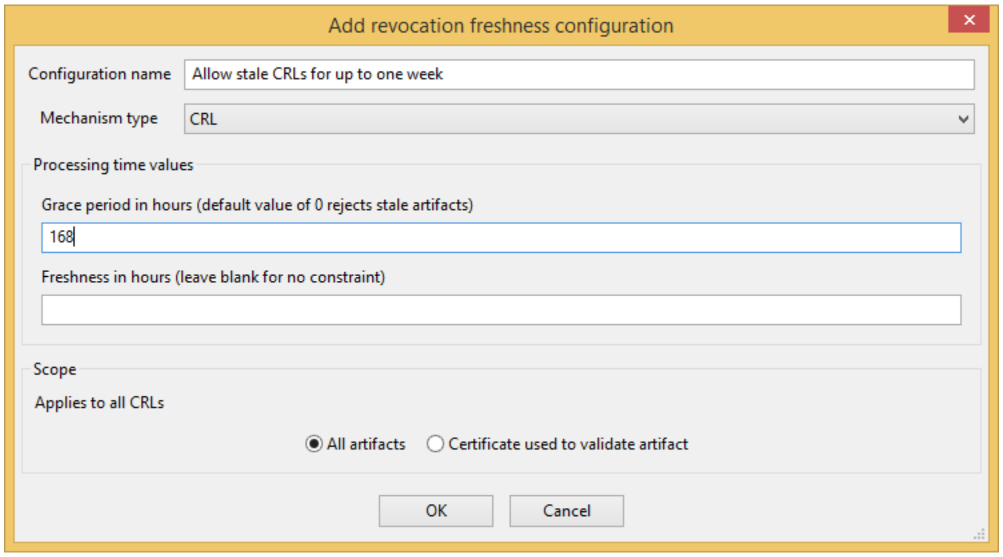
☑ Check OCSP AIA ☑ Check CRLs ☑ Check CRL DP - HTTP ☐ Check CRL DP - LDAP ☑ Only apply CRL grace periods as a last resort

CRL folder: E:\PITT_testing\Artifacts\VeryOldCrls

OCSP AIA Nonce: Do not send nonce

Figure 21 Revocation settings

To add a revocation freshness configuration click the **Add configuration** button below the **Revocation freshness configuration** list. A dialog like the one below will be displayed. Enter a configuration name and select a mechanism (only CRLs are supported in this release). Enter a **Grace period in hours** to allow use of stale CRLs. The grace period defines the maximum allowable difference between the current time and the nextUpdate value in a CRL. Enter a **Freshness in hours** to disallow revocation information that was generated too far in the past (even if not stale). The freshness value defines the maximum allowable difference between the current time and the thisUpdate value in a CRL. The scope of the revocation freshness configuration can be set to all CRLs or limited by CA certificate. For example, to allow stale CRLs only for a given CA set the configurations as described above then click the **Certificate used to validate artifact** radio button and browse to the CA certificate.



Configuration name: Allow stale CRLs for up to one week

Mechanism type: CRL

Grace period in hours (default value of 0 rejects stale artifacts): 168

Freshness in hours (leave blank for no constraint):

Applies to all CRLs

[✓ All artifacts] Certificate used to validate artifact

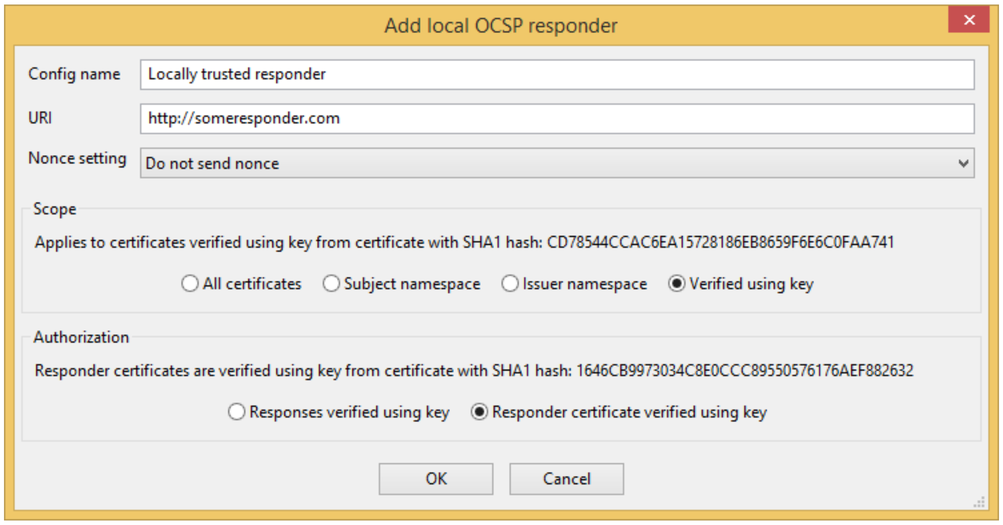
Figure 22 Revocation freshness configuration

In addition to the mechanisms described above (i.e., **CRL folder**, **Check OCSP AIA** and etc.), three additional mechanisms may be defined: **Locally trusted OCSP**, **Blacklist** and **No check**. The **Blacklist** mechanism simply lists certificates that are marked as revoked independent of and without consulting any OCSP or CRL sources. The No check option is essentially a whitelist. Certificates included listed in the **No check** table are marked as not revoked independent of and without consulting any OCSP or CRL sources.

To configure a locally trusted OCSP responder, choose **Locally trusted OCSP** option in the **Type** field in the **Mechanisms** box then click the **Add responder** button. A dialog box similar to the dialog shown below will be displayed. Enter a name for the configuration in the **Config name** field and enter the URI where the OCSP responder can be reached in the **URI** field then select a **Nonce setting**. When **Do not send nonce** is selected, requests sent to this responder per this configuration will not include a nonce. When either **Send nonce and require match in response** or **Send nonce but do not require match in response** is selected, a nonce will be included in requests with response processing proceeding per the selected option. The **Send nonce but do not require match in response** option is primarily useful when the responder does or may return cached responses. If a freshly generated response is required, use the **Send nonce and require match in response** option, keeping in mind that the responder may not comply and responses that do not match will be discarded and not be used for revocation status determination.

The **Scope** option can be used to limit the applicability of the configuration. When **All certificates** is selected, the configuration may be used to determine the revocation status of any certificate. This is not typically used. The **Subject namespace** and **Issuer namespace** options can be used to limit applicability to certificates with a subject or issuer name subordinate to the configured name. The **Verified using key** option limits the applicability to certificates verified using the public key in the specified certificate.

The **Authorization** setting determines how the responder is authorized. When the **Responses verified using key** option is used, responses retrieved per this configuration must be verified using the public key in the specified certificate. When the **Responder certificate verified using key** option is selected, the response must be verified using a responder certificate that can be verified using the public key in the specified certificate.


Add local OCSP responder

Config name: Locally trusted responder

URI: http://someresponder.com

Nonce setting: Do not send nonce

Scope

Applies to certificates verified using key from certificate with SHA1 hash: CD78544CCAC6EA15728186EB8659F6E6C0FAA741

All certificates Subject namespace Issuer namespace [✓ Verified using key]

Authorization

Responder certificates are verified using key from certificate with SHA1 hash: 1646CB9973034C8E0CCC89550576176AEF882632

Responses verified using key [✓ Responder certificate verified using key]

Figure 23 Local OCSP responder configuration

To add a certificate to the blacklist, select **Blacklist** in the **Type** field then click the **Add certificate to blacklist** option. To remove a certificate from the list, select the certificate and click the **Remove certificate from blacklist**.
To add a certificate to the no check list, select **No check** in the **Type** field then click the **Add certificate** button. A dialog similar to the dialog shown below will be displayed. Browse to the certificate then choose whether to add only the specified certificate or also certificates subordinate to this certificate. Note the whitelist option is certificate based, not public key based. Thus if multiple certificates are issued for the same key and all should be whitelisted, each must be added.

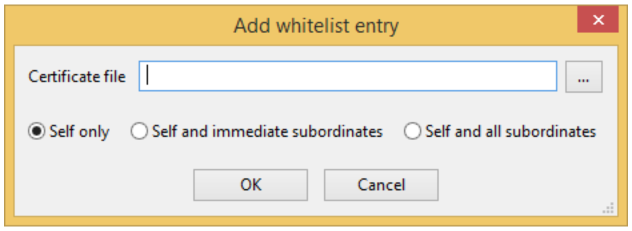
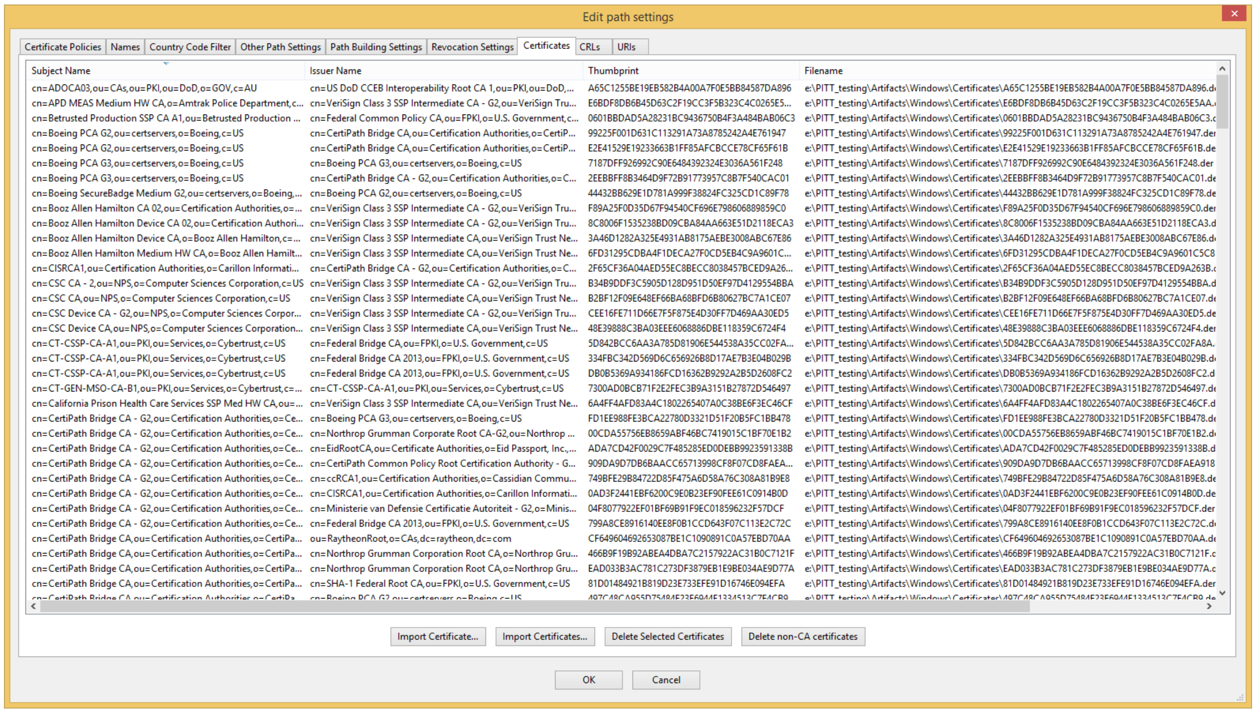


Figure 24 Whitelist entry configuration

Certificates

The **Certificates** tab displays a list of certificate files present in the **Certs folder** specified on the **Path Building Settings** tab. The list displays the subject name, issuer name, the SHA1 hash of the DER encoded certificate and filename for each certificate. The list can be sorted on each column in either ascending or descending order by clicking the column label. To import a certificate or folder of certificates, click the **Import Certificate** button or **Import Certificates** button. To delete a certificate file, select the certificate to delete then click the **Delete Selected Certificates** button. To delete non-CA certificates, click the **Delete non-CA certificates** button. The list is populated when the **Edit path settings** notebook is displayed. After changing the **Certs folder** setting, dismiss and re-launch the **Edit path settings** notebook to refresh the **Certificates** list.

Care should be exercised when deleting certificate files that were downloaded automatically (i.e., the files named using a hash). The **URIs** tab displays the last modified time retrieved from the server when a certificate (or PKCS #7 or CRL) file was downloaded. If the local copy of the file is deleted and the URI is not deleted from the URI database, a fresh copy of the certificate may not be re-downloaded.



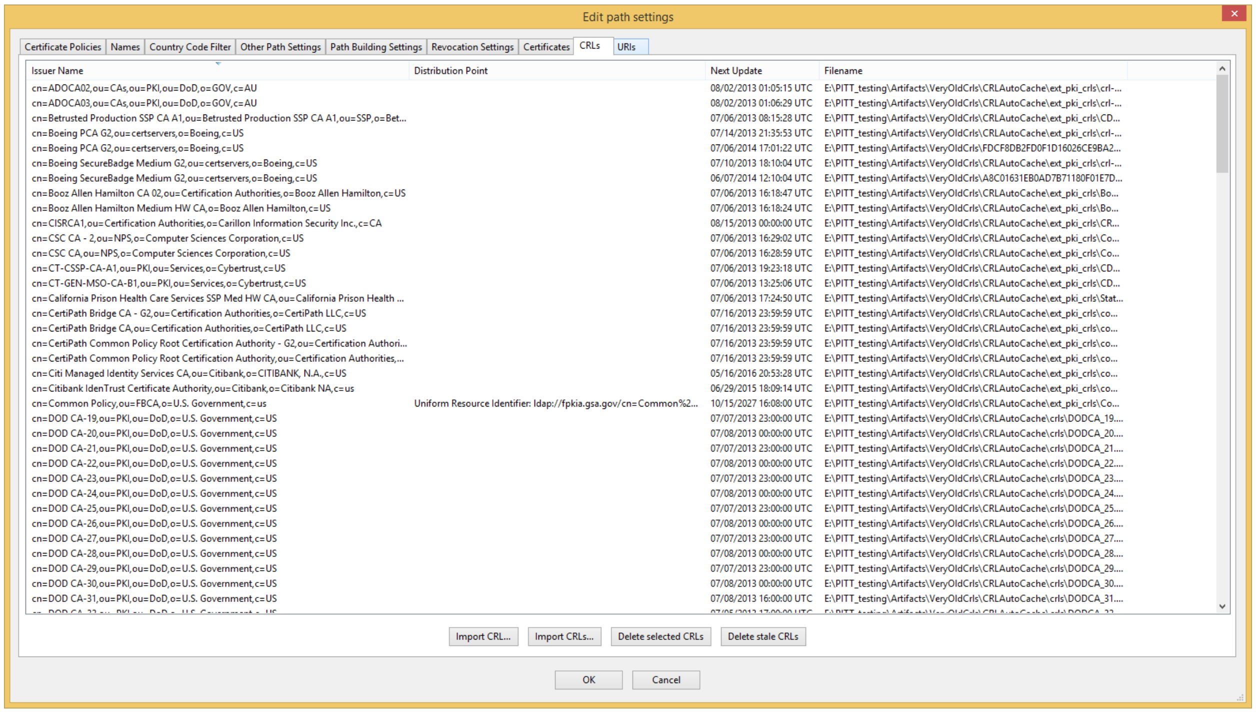
Subject Name	Issuer Name	Thumbprint	Filename
cn=ADOCA03,ou=CAs,ou=PKI,ou=DoD,o=GOV,c=AU	cn=US DoD CCEB Interoperability Root CA 1,ou=PKI,ou=DoD,...	A65C1255BE19EB582B4A00A7F0E5BB84587DA896	e:\PITT_testing\Artifacts\Windows\Certificates\A65C1255BE19EB582B4A00A7F0E5BB84587DA896.d
cn=APD MEAS Medium HW CA,o=Amtrak Police Department,c...	cn=VeriSign Class 3 SSP Intermediate CA - G2,ou=VeriSign Tru...	E6BDF8DB6B45D63C2F19CC3F5B323C4C0265E5...	e:\PITT_testing\Artifacts\Windows\Certificates\E6BDF8DB6B45D63C2F19CC3F5B323C4C0265E5AA.
cn=Betrusted Production SSP CA A1,ou=Betrusted Production ...	cn=Federal Common Policy CA,ou=FPKI,o=U.S. Government,c...	0601BBDAD5A28231BC9436750B4F3A484BAB06C3	e:\PITT_testing\Artifacts\Windows\Certificates\0601BBDAD5A28231BC9436750B4F3A484BAB06C3.
cn=Boeing PCA G2,ou=certservers,o=Boeing,c=US	cn=CertiPath Bridge CA,ou=Certification Authorities,o=CertiP...	99225F001D631C113291A73A8785242A4E761947	e:\PITT_testing\Artifacts\Windows\Certificates\99225F001D631C113291A73A8785242A4E761947.der
cn=Boeing PCA G2,ou=certservers,o=Boeing,c=US	cn=CertiPath Bridge CA,ou=Certification Authorities,o=CertiP...	E2E41529E19233663B1FF85AFCBCCE78CF65F61B	e:\PITT_testing\Artifacts\Windows\Certificates\E2E41529E19233663B1FF85AFCBCCE78CF65F61B.de
cn=Boeing PCA G3,ou=certservers,o=Boeing,c=US	cn=Boeing PCA G3,ou=certservers,o=Boeing,c=US	7187DFF926992C90E6484392324E3036A561F248	e:\PITT_testing\Artifacts\Windows\Certificates\7187DFF926992C90E6484392324E3036A561F248.der
cn=Boeing PCA G3,ou=certservers,o=Boeing,c=US	cn=CertiPath Bridge CA - G2,ou=Certification Authorities,o=C...	2EEBBFF8B3464D9F72B91773957C8B7F540CAC01	e:\PITT_testing\Artifacts\Windows\Certificates\2EEBBFF8B3464D9F72B91773957C8B7F540CAC01.de
cn=Boeing SecureBadge Medium G2,ou=certservers,o=Boeing,...	cn=Boeing PCA G2,ou=certservers,o=Boeing,c=US	44432BB629E1D781A999F38824FC325CD1C89F78	e:\PITT_testing\Artifacts\Windows\Certificates\44432BB629E1D781A999F38824FC325CD1C89F78.de
cn=Booz Allen Hamilton CA 02,ou=Certification Authorities,o=...	cn=VeriSign Class 3 SSP Intermediate CA - G2,ou=VeriSign Tru...	F89A25F0D35D67F94540CF696E798606889859C0	e:\PITT_testing\Artifacts\Windows\Certificates\F89A25F0D35D67F94540CF696E798606889859C0.der
cn=Booz Allen Hamilton Device CA 02,ou=Certification Authori...	cn=VeriSign Class 3 SSP Intermediate CA - G2,ou=VeriSign Tru...	8C8006F1535238BD09CBA84AA663E51D2118ECA3	e:\PITT_testing\Artifacts\Windows\Certificates\8C8006F1535238BD09CBA84AA663E51D2118ECA3.d
cn=Booz Allen Hamilton Device CA,o=Booz Allen Hamilton,c=...	cn=VeriSign Class 3 SSP Intermediate CA,ou=VeriSign Trust Ne...	3A46D1282A325E4931AB8175AEBE3008ABC67E86	e:\PITT_testing\Artifacts\Windows\Certificates\3A46D1282A325E4931AB8175AEBE3008ABC67E86.d
cn=Booz Allen Hamilton Medium HW CA,o=Booz Allen Hamilt...	cn=VeriSign Class 3 SSP Intermediate CA,ou=VeriSign Trust Ne...	6FD31295CDBA4F1DECA27F0CD5EB4C9A9601C...	e:\PITT_testing\Artifacts\Windows\Certificates\6FD31295CDBA4F1DECA27F0CD5EB4C9A9601C5C8
cn=CISRCA1,ou=Certification Authorities,o=Carillon Informati...	cn=CertiPath Bridge CA - G2,ou=Certification Authorities,o=C...	2F65CF36A04AED55EC8BECC8038457BCED9A26...	e:\PITT_testing\Artifacts\Windows\Certificates\2F65CF36A04AED55EC8BECC8038457BCED9A263B.
cn=CSC CA - 2,ou=NPS,o=Computer Sciences Corporation,c=US	cn=VeriSign Class 3 SSP Intermediate CA - G2,ou=VeriSign Tru...	B34B9DDF3C5905D128D951D50EF97D4129554BBA	e:\PITT_testing\Artifacts\Windows\Certificates\B34B9DDF3C5905D128D951D50EF97D4129554BBA.d
cn=CSC CA,ou=NPS,o=Computer Sciences Corporation,c=US	cn=VeriSign Class 3 SSP Intermediate CA,ou=VeriSign Trust Ne...	B2BF12F09E648EF66BA68BFD6B80627BC7A1CE07	e:\PITT_testing\Artifacts\Windows\Certificates\B2BF12F09E648EF66BA68BFD6B80627BC7A1CE07.de
cn=CSC Device CA - G2,ou=NPS,o=Computer Sciences Corpor...	cn=VeriSign Class 3 SSP Intermediate CA - G2,ou=VeriSign Tru...	CEE16FE711D66E7F5F875E4D30FF7D469AA30ED5	e:\PITT_testing\Artifacts\Windows\Certificates\CEE16FE711D66E7F5F875E4D30FF7D469AA30ED5.de
cn=CSC Device CA,ou=NPS,o=Computer Sciences Corporation...	cn=VeriSign Class 3 SSP Intermediate CA,ou=VeriSign Trust Ne...	48E39888C3BA03EEE6068886DBE118359C6724F4	e:\PITT_testing\Artifacts\Windows\Certificates\48E39888C3BA03EEE6068886DBE118359C6724F4.der
cn=CT-CSSP-CA-A1,ou=PKI,ou=Services,o=Cybertrust,c=US	cn=Federal Bridge CA,ou=FPKI,o=U.S. Government,c=US	5D842BCC6AA3A785D81906E544538A35CC02FA...	e:\PITT_testing\Artifacts\Windows\Certificates\5D842BCC6AA3A785D81906E544538A35CC02FA8A.
cn=CT-CSSP-CA-A1,ou=PKI,ou=Services,o=Cybertrust,c=US	cn=Federal Bridge CA 2013,ou=FPKI,o=U.S. Government,c=US	334FBC342D569D6C656926B8D17AE7B3E04B029B	e:\PITT_testing\Artifacts\Windows\Certificates\334FBC342D569D6C656926B8D17AE7B3E04B029B.d
cn=CT-CSSP-CA-A1,ou=PKI,ou=Services,o=Cybertrust,c=US	cn=Federal Bridge CA 2013,ou=FPKI,o=U.S. Government,c=US	DB0B5369A934186FCD16362B9292A2B5D2608FC2	e:\PITT_testing\Artifacts\Windows\Certificates\DB0B5369A934186FCD16362B9292A2B5D2608FC2.d
cn=CT-GEN-MSO-CA-B1,ou=PKI,ou=Services,o=Cybertrust,c=...	cn=CT-CSSP-CA-A1,ou=PKI,ou=Services,o=Cybertrust,c=US	7300AD0BCB71F2E2FEC3B9A3151B27872D546497	e:\PITT_testing\Artifacts\Windows\Certificates\7300AD0BCB71F2E2FEC3B9A3151B27872D546497.de
cn=California Prison Health Care Services SSP Med HW CA,ou=...	cn=VeriSign Class 3 SSP Intermediate CA,ou=VeriSign Trust Ne...	6A4FF4AFD83A4C1802265407A0C38BE6F3EC46CF	e:\PITT_testing\Artifacts\Windows\Certificates\6A4FF4AFD83A4C1802265407A0C38BE6F3EC46CF.d
cn=CertiPath Bridge CA - G2,ou=Certification Authorities,o=Ce...	cn=Boeing PCA G3,ou=certservers,o=Boeing,c=US	FD1EE988FE3BCA22780D3321D51F20B5FC1BB478	e:\PITT_testing\Artifacts\Windows\Certificates\FD1EE988FE3BCA22780D3321D51F20B5FC1BB478.de
cn=CertiPath Bridge CA - G2,ou=Certification Authorities,o=Ce...	cn=Northrop Grumman Corporate Root CA-G2,ou=Northrop ...	00CDA55756EB8659ABF46BC7419015C1BF70E1B2	e:\PITT_testing\Artifacts\Windows\Certificates\00CDA55756EB8659ABF46BC7419015C1BF70E1B2.d
cn=CertiPath Bridge CA - G2,ou=Certification Authorities,o=Ce...	cn=EidRootCA,ou=Certificate Authorities,o=Eid Passport, Inc.,...	ADA7CD42F0029C7F485285ED0DEBB923591338B	e:\PITT_testing\Artifacts\Windows\Certificates\ADA7CD42F0029C7F485285ED0DEBB923591338B.d
cn=CertiPath Bridge CA - G2,ou=Certification Authorities,o=Ce...	cn=CertiPath Common Policy Root Certification Authority - G...	909DA9D7DB6BAACC65713998CF8F07CD8FAEA...	e:\PITT_testing\Artifacts\Windows\Certificates\909DA9D7DB6BAACC65713998CF8F07CD8FAEA918
cn=CertiPath Bridge CA - G2,ou=Certification Authorities,o=Ce...	cn=ccRCA1,ou=Certification Authorities,o=Cassidian Commu...	749BFE29B84722D85F475A6D58A76C308A81B9E8	e:\PITT_testing\Artifacts\Windows\Certificates\749BFE29B84722D85F475A6D58A76C308A81B9E8.de
cn=CertiPath Bridge CA - G2,ou=Certification Authorities,o=Ce...	cn=CISRCA1,ou=Certification Authorities,o=Carillon Informati...	0AD3F2441EBF6200C9E0B23EF90FEE61C0914B0D	e:\PITT_testing\Artifacts\Windows\Certificates\0AD3F2441EBF6200C9E0B23EF90FEE61C0914B0D.de
cn=CertiPath Bridge CA - G2,ou=Certification Authorities,o=Ce...	cn=Ministerie van Defensie Certificatie Autoriteit - G2,o=Minis...	04F8077922EF01BF69B91F9EC018596232F57DCF	e:\PITT_testing\Artifacts\Windows\Certificates\04F8077922EF01BF69B91F9EC018596232F57DCF.der
cn=CertiPath Bridge CA - G2,ou=Certification Authorities,o=Ce...	cn=Federal Bridge CA 2013,ou=FPKI,o=U.S. Government,c=US	799A8CE8916140EE8F0B1CCD643F07C113E2C72C	e:\PITT_testing\Artifacts\Windows\Certificates\799A8CE8916140EE8F0B1CCD643F07C113E2C72C.d
cn=CertiPath Bridge CA,ou=Certification Authorities,o=CertiPa...	ou=RaytheonRoot,o=CAs,dc=raytheon,dc=com	CF649604692653087BE1C1090891C0A57EBD70AA	e:\PITT_testing\Artifacts\Windows\Certificates\CF649604692653087BE1C1090891C0A57EBD70AA.de
cn=CertiPath Bridge CA,ou=Certification Authorities,o=CertiPa...	cn=Northrop Grumman Corporation Root CA,o=Northrop Gru...	466B9F19B92ABEA4DBA7C2157922AC31B0C7121F	e:\PITT_testing\Artifacts\Windows\Certificates\466B9F19B92ABEA4DBA7C2157922AC31B0C7121F.
cn=CertiPath Bridge CA,ou=Certification Authorities,o=CertiPa...	cn=Northrop Grumman Corporation Root CA,o=Northrop Gru...	EAD033B3AC781C273DF3879EB1E9BE034AE9D77A	e:\PITT_testing\Artifacts\Windows\Certificates\EAD033B3AC781C273DF3879EB1E9BE034AE9D77A.
cn=CertiPath Bridge CA,ou=Certification Authorities,o=CertiPa...	cn=SHA-1 Federal Root CA,ou=FPKI,o=U.S. Government,c=US	81D01484921B819D23E733EFE91D16746E094EFA	e:\PITT_testing\Artifacts\Windows\Certificates\81D01484921B819D23E733EFE91D16746E094EFA.der
cn=CertiPath Bridge CA,ou=Certification Authorities,o=CertiPa...	cn=Boeing PCA G2,ou=certservers,o=Boeing,c=US	497C48CA955D75484F23F6944E1334513C7F4CB9	e:\PITT_testing\Artifacts\Windows\Certificates\497C48CA955D75484F23F6944E1334513C7F4CB9.de

Figure 25 Certificates list

CRLs

The **CRLs** tab displays a list of CRL files present in the **CRLs folder** specified on the **Revocation Settings** tab. The list displays the issuer name, distribution point, next update and filename for each CRL. The list can be sorted on each column in either ascending or descending order by clicking the column label. To import a CRL or folder of CRLs, click the **Import CRL** button or **Import CRL** button. To delete a CRL file, select the CRL to delete then click the **Delete Selected CRLs** button. To delete stale CRLs, click the **Delete stale CRLs** button. The list is populated when the **Edit path settings** notebook is displayed. After changing the **CRLs folder** setting, dismiss and re-launch the **Edit path settings** notebook to refresh the **CRLs** list.

Care should be exercised when deleting CRL files that were downloaded automatically (i.e., the files named using a hash). The **URIs** tab displays the last modified time retrieved from the server when a CRL (or PKCS #7 or certificate) file was downloaded. If the local copy of the file is deleted and the URI is not deleted from the URI database, a fresh copy of the certificate may not be re-downloaded.

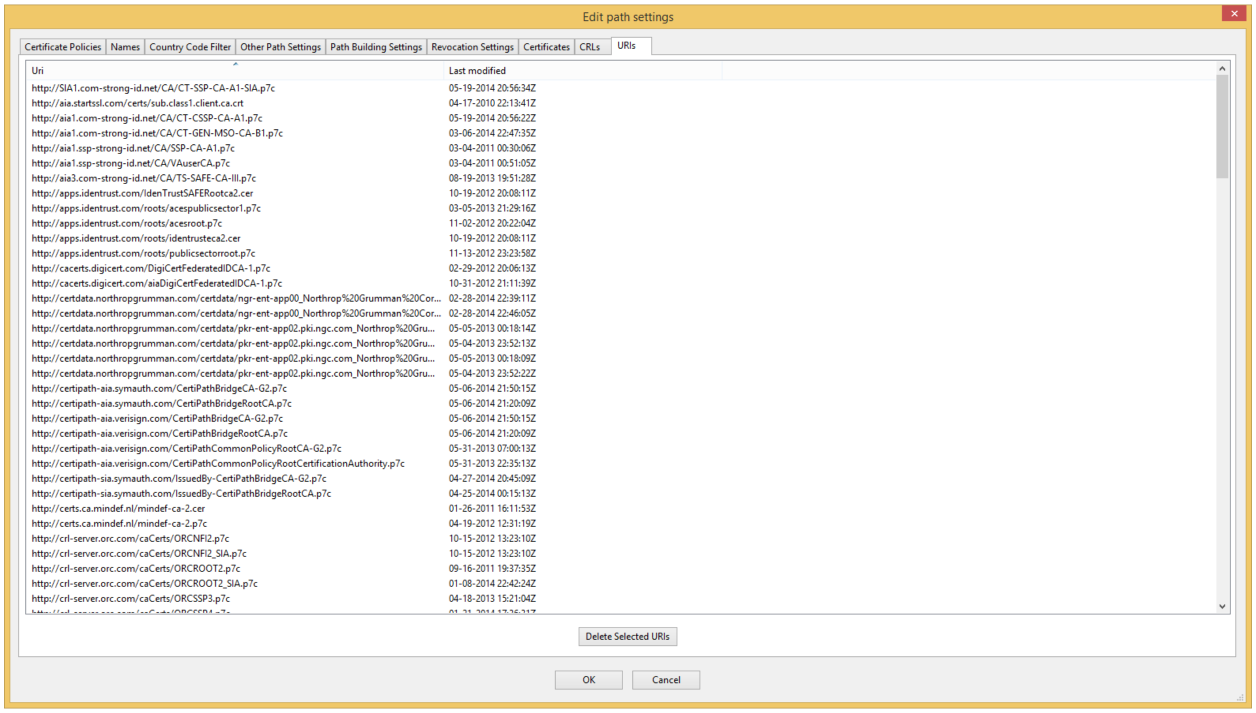


Issuer Name	Distribution Point	Next Update	Filename
cn=ADOCA02,ou=CAs,ou=PKI,ou=DoD,o=GOV,c=AU		08/02/2013 01:05:15 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\crl-...
cn=ADOCA03,ou=CAs,ou=PKI,ou=DoD,o=GOV,c=AU		08/02/2013 01:06:29 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\crl-...
cn=Betrusted Production SSP CA A1,ou=Betrusted Production SSP CA A1,ou=SSP,o=Bet...		07/06/2013 08:15:28 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\CD...
cn=Boeing PCA G2,ou=certservers,o=Boeing,c=US		07/14/2013 21:35:53 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\crl-...
cn=Boeing PCA G2,ou=certservers,o=Boeing,c=US		07/06/2014 17:01:22 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\FDCF8DB2FD0F1D16026CE9BA2...
cn=Boeing SecureBadge Medium G2,ou=certservers,o=Boeing,c=US		07/10/2013 18:10:04 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\crl-...
cn=Boeing SecureBadge Medium G2,ou=certservers,o=Boeing,c=US		06/07/2014 12:10:04 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\A8C01631EB0AD7B71180F01E7D...
cn=Booz Allen Hamilton CA 02,ou=Certification Authorities,o=Booz Allen Hamilton,c=US		07/06/2013 16:18:47 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\Bo...
cn=Booz Allen Hamilton Medium HW CA,o=Booz Allen Hamilton,c=US		07/06/2013 16:18:24 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\Bo...
cn=CISRCA1,ou=Certification Authorities,o=Carillon Information Security Inc.,c=CA		08/15/2013 00:00:00 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\CR...
cn=CSC CA - 2,ou=NPS,o=Computer Sciences Corporation,c=US		07/06/2013 16:29:02 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\Co...
cn=CSC CA,ou=NPS,o=Computer Sciences Corporation,c=US		07/06/2013 16:28:59 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\Co...
cn=CT-CSSP-CA-A1,ou=PKI,ou=Services,o=Cybertrust,c=US		07/06/2013 19:23:18 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\CD...
cn=CT-GEN-MSO-CA-B1,ou=PKI,ou=Services,o=Cybertrust,c=US		07/06/2013 13:25:06 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\CD...
cn=California Prison Health Care Services SSP Med HW CA,ou=California Prison Health ...		07/06/2013 17:24:50 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\Stat...
cn=CertiPath Bridge CA - G2,ou=Certification Authorities,o=CertiPath LLC,c=US		07/16/2013 23:59:59 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\co...
cn=CertiPath Bridge CA,ou=Certification Authorities,o=CertiPath LLC,c=US		07/16/2013 23:59:59 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\co...
cn=CertiPath Common Policy Root Certification Authority - G2,ou=Certification Authori...		07/16/2013 23:59:59 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\co...
cn=CertiPath Common Policy Root Certification Authority,ou=Certification Authorities,...		07/16/2013 23:59:59 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\co...
cn=Citi Managed Identity Services CA,ou=Citibank,o=CITIBANK, N.A.,c=US		05/16/2016 20:53:28 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\co...
cn=Citibank IdenTrust Certificate Authority,ou=Citibank,o=Citibank NA,c=us		06/29/2015 18:09:14 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\co...
cn=Common Policy,ou=FBCA,o=U.S. Government,c=us	Uniform Resource Identifier: ldap://fpkia.gsa.gov/cn=Common%2...	10/15/2027 16:08:00 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\ext_pki_crls\Co...
cn=DOD CA-19,ou=PKI,ou=DoD,o=U.S. Government,c=US		07/07/2013 23:00:00 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\crls\DODCA_19....
cn=DOD CA-20,ou=PKI,ou=DoD,o=U.S. Government,c=US		07/08/2013 00:00:00 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\crls\DODCA_20....
cn=DOD CA-21,ou=PKI,ou=DoD,o=U.S. Government,c=US		07/07/2013 23:00:00 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\crls\DODCA_21....
cn=DOD CA-22,ou=PKI,ou=DoD,o=U.S. Government,c=US		07/08/2013 00:00:00 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\crls\DODCA_22....
cn=DOD CA-23,ou=PKI,ou=DoD,o=U.S. Government,c=US		07/07/2013 23:00:00 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\crls\DODCA_23....
cn=DOD CA-24,ou=PKI,ou=DoD,o=U.S. Government,c=US		07/08/2013 00:00:00 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\crls\DODCA_24....
cn=DOD CA-25,ou=PKI,ou=DoD,o=U.S. Government,c=US		07/07/2013 23:00:00 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\crls\DODCA_25....
cn=DOD CA-26,ou=PKI,ou=DoD,o=U.S. Government,c=US		07/08/2013 00:00:00 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\crls\DODCA_26....
cn=DOD CA-27,ou=PKI,ou=DoD,o=U.S. Government,c=US		07/07/2013 23:00:00 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\crls\DODCA_27....
cn=DOD CA-28,ou=PKI,ou=DoD,o=U.S. Government,c=US		07/08/2013 00:00:00 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\crls\DODCA_28....
cn=DOD CA-29,ou=PKI,ou=DoD,o=U.S. Government,c=US		07/07/2013 23:00:00 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\crls\DODCA_29....
cn=DOD CA-30,ou=PKI,ou=DoD,o=U.S. Government,c=US		07/08/2013 00:00:00 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\crls\DODCA_30....
cn=DOD CA-31,ou=PKI,ou=DoD,o=U.S. Government,c=US		07/08/2013 16:00:00 UTC	E:\PITT_testing\Artifacts\VeryOldCrls\CRLAutoCache\crls\DODCA_31....

Figure 26 CRLs

URIs

The **URIs** tab displays a list of URIs read from the uriLastModified.db file present in the **Certs folder** and **CRLs folder** locations. The list includes the URI and last modified time returned from the server. The last modified value is used to avoid downloading artifacts that have not changed since a previous download operation. As noted above, when manually deleting automatically downloaded certificates or CRL files, the uriLastModified.db file should either be deleted or the URI associated with the manually deleted file should be deleted from the uriLastModified.db list.



Uri	Last modified
http://SIA1.com-strong-id.net/CA/CT-SSP-CA-A1-SIA.p7c	05-19-2014 20:56:34Z
http://aia.startssl.com/certs/sub.class1.client.ca.crt	04-17-2010 22:13:41Z
http://aia1.com-strong-id.net/CA/CT-CSSP-CA-A1.p7c	05-19-2014 20:56:22Z
http://aia1.com-strong-id.net/CA/CT-GEN-MSO-CA-B1.p7c	03-06-2014 22:47:35Z
http://aia1.ssp-strong-id.net/CA/SSP-CA-A1.p7c	03-04-2011 00:30:06Z
http://aia1.ssp-strong-id.net/CA/VAuserCA.p7c	03-04-2011 00:51:05Z
http://aia3.com-strong-id.net/CA/TS-SAFE-CA-III.p7c	08-19-2013 19:51:28Z
http://apps.identrust.com/IdenTrustSAFERootca2.cer	10-19-2012 20:08:11Z
http://apps.identrust.com/roots/acespublicsector1.p7c	03-05-2013 21:29:16Z
http://apps.identrust.com/roots/acesroot.p7c	11-02-2012 20:22:04Z
http://apps.identrust.com/roots/identrusteca2.cer	10-19-2012 20:08:11Z
http://apps.identrust.com/roots/publicsectorroot.p7c	11-13-2012 23:23:58Z
http://cacerts.digicert.com/DigiCertFederatedIDCA-1.p7c	02-29-2012 20:06:13Z
http://cacerts.digicert.com/aiaDigiCertFederatedIDCA-1.p7c	10-31-2012 21:11:39Z
http://certdata.northropgrumman.com/certdata/ngr-ent-app00_Northrop%20Grumman%20Cor...	02-28-2014 22:39:11Z
http://certdata.northropgrumman.com/certdata/ngr-ent-app00_Northrop%20Grumman%20Cor...	02-28-2014 22:46:05Z
http://certdata.northropgrumman.com/certdata/pkr-ent-app02.pki.ngc.com_Northrop%20Gru...	05-05-2013 00:18:14Z
http://certdata.northropgrumman.com/certdata/pkr-ent-app02.pki.ngc.com_Northrop%20Gru...	05-04-2013 23:52:13Z
http://certdata.northropgrumman.com/certdata/pkr-ent-app02.pki.ngc.com_Northrop%20Gru...	05-05-2013 00:18:09Z
http://certdata.northropgrumman.com/certdata/pkr-ent-app02.pki.ngc.com_Northrop%20Gru...	05-04-2013 23:52:22Z
http://certipath-aia.symauth.com/CertiPathBridgeCA-G2.p7c	05-06-2014 21:50:15Z
http://certipath-aia.symauth.com/CertiPathBridgeRootCA.p7c	05-06-2014 21:20:09Z
http://certipath-aia.verisign.com/CertiPathBridgeCA-G2.p7c	05-06-2014 21:50:15Z
http://certipath-aia.verisign.com/CertiPathBridgeRootCA.p7c	05-06-2014 21:20:09Z
http://certipath-aia.verisign.com/CertiPathCommonPolicyRootCA-G2.p7c	05-31-2013 07:00:13Z
http://certipath-aia.verisign.com/CertiPathCommonPolicyRootCertificationAuthority.p7c	05-31-2013 22:35:13Z
http://certipath-sia.symauth.com/IssuedBy-CertiPathBridgeCA-G2.p7c	04-27-2014 20:45:09Z
http://certipath-sia.symauth.com/IssuedBy-CertiPathBridgeRootCA.p7c	04-25-2014 00:15:13Z
http://certs.ca.mindef.nl/mindef-ca-2.cer	01-26-2011 16:11:53Z
http://certs.ca.mindef.nl/mindef-ca-2.p7c	04-19-2012 12:31:19Z
http://crl-server.orc.com/caCerts/ORCNFI2.p7c	10-15-2012 13:23:10Z
http://crl-server.orc.com/caCerts/ORCNFI2_SIA.p7c	10-15-2012 13:23:10Z
http://crl-server.orc.com/caCerts/ORCROOT2.p7c	09-16-2011 19:37:35Z
http://crl-server.orc.com/caCerts/ORCROOT2_SIA.p7c	01-08-2014 22:42:24Z
http://crl-server.orc.com/caCerts/ORCSSP3.p7c	04-18-2013 15:21:04Z

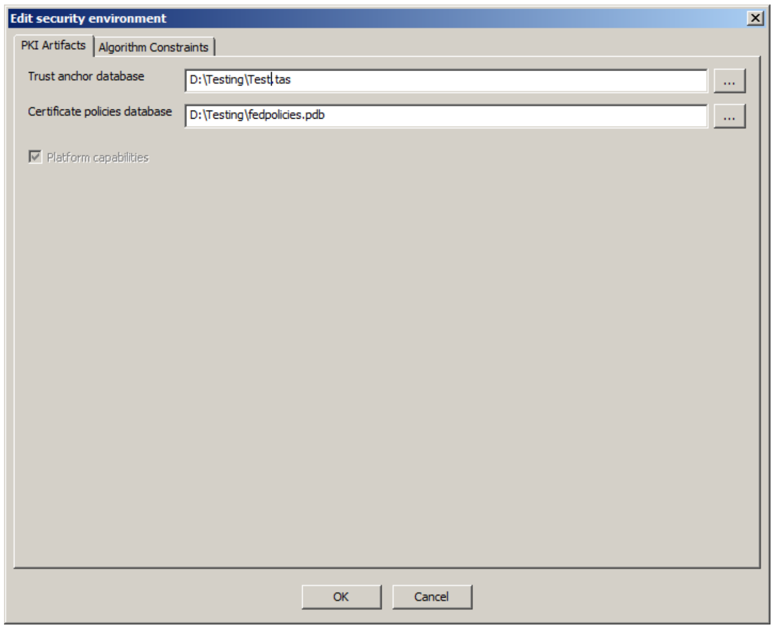
Figure 27 Last modified times for URIs

Edit security environments

Security environments contain settings that are less volatile. These settings include:

- TACT trust anchor store to use during certification path validation
- Certificate policies database to use during path settings configuration and logging of certification path validation results
- Cryptographic algorithm prohibitions

To specify a TACT trust anchor store or certificate policies database, use the **PKI Artifacts** tab in the security environment editor dialog, as shown below. To specify a TA store or policies database, click the **...** button adjacent to the corresponding field and navigate to the desired file. On Windows, TA stores are typically stored in `C:\TactSettings\Tas` and policies databases are stored in `C:\TactSettings`. On Linux, the usual locations are `/etc/tact/tas` and `/etc/tact`, respectively.


Edit security environment

PKI Artifacts | Algorithm Constraints

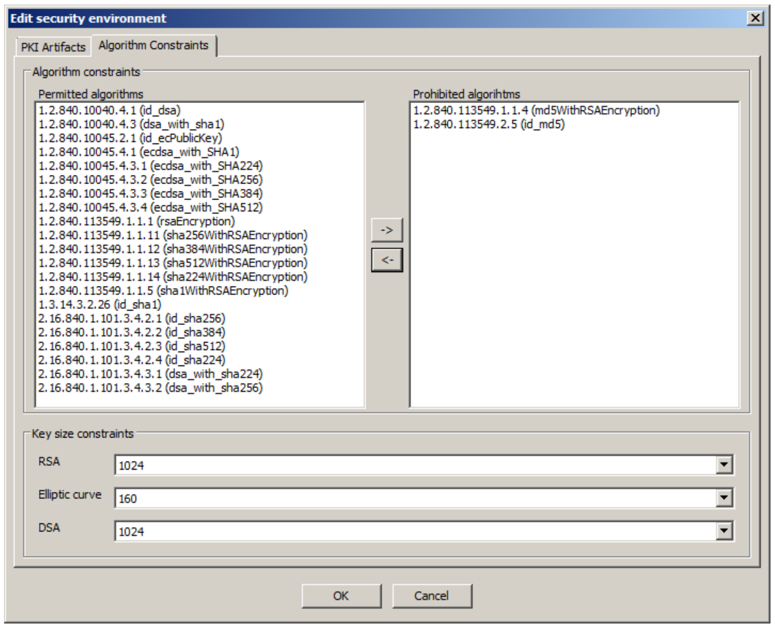
Trust anchor database: D:\Testing\Test.tas

Certificate policies database: D:\Testing\fedpolicies.pdb

☑ Platform capabilities

Figure 28 TA store and certificate policies database specification

To specific cryptographic algorithm prohibitions or to define minimum acceptable cryptographic key sizes, use the **Algorithm Constraints** tab as shown below. To prohibit an algorithm, select the algorithm in the left list box and click the –> button (or double click the algorithm). To remove an algorithm prohibition, select the algorithm in the right list box and click the <– button (or double click the algorithm). To specify a minimum key size, choose a value from the drop list corresponding to the algorithm of interest.


Edit security environment

PKI Artifacts | Algorithm Constraints

Algorithm constraints

Permitted algorithms

```
1.2.840.10040.4.1 (id_dsa)
1.2.840.10040.4.3 (dsa_with_sha1)
1.2.840.10045.2.1 (id_ecPublicKey)
1.2.840.10045.4.1 (ecdsa_with_SHA1)
1.2.840.10045.4.3.1 (ecdsa_with_SHA224)
1.2.840.10045.4.3.2 (ecdsa_with_SHA256)
1.2.840.10045.4.3.3 (ecdsa_with_SHA384)
1.2.840.10045.4.3.4 (ecdsa_with_SHA512)
1.2.840.113549.1.1.1 (rsaEncryption)
1.2.840.113549.1.1.11 (sha256WithRSAEncryption)
1.2.840.113549.1.1.12 (sha384WithRSAEncryption)
1.2.840.113549.1.1.13 (sha512WithRSAEncryption)
1.2.840.113549.1.1.14 (sha224WithRSAEncryption)
1.2.840.113549.1.1.5 (sha1WithRSAEncryption)
1.3.14.3.2.26 (id_sha1)
2.16.840.1.101.3.4.2.1 (id_sha256)
2.16.840.1.101.3.4.2.2 (id_sha384)
2.16.840.1.101.3.4.2.3 (id_sha512)
2.16.840.1.101.3.4.2.4 (id_sha224)
2.16.840.1.101.3.4.3.1 (dsa_with_sha224)
2.16.840.1.101.3.4.3.2 (dsa_with_sha256)
```

Prohibited algorihtms

```
1.2.840.113549.1.1.4 (md5WithRSAEncryption)
1.2.840.113549.2.5 (id_md5)
```

Key size constraints

RSA: 1024

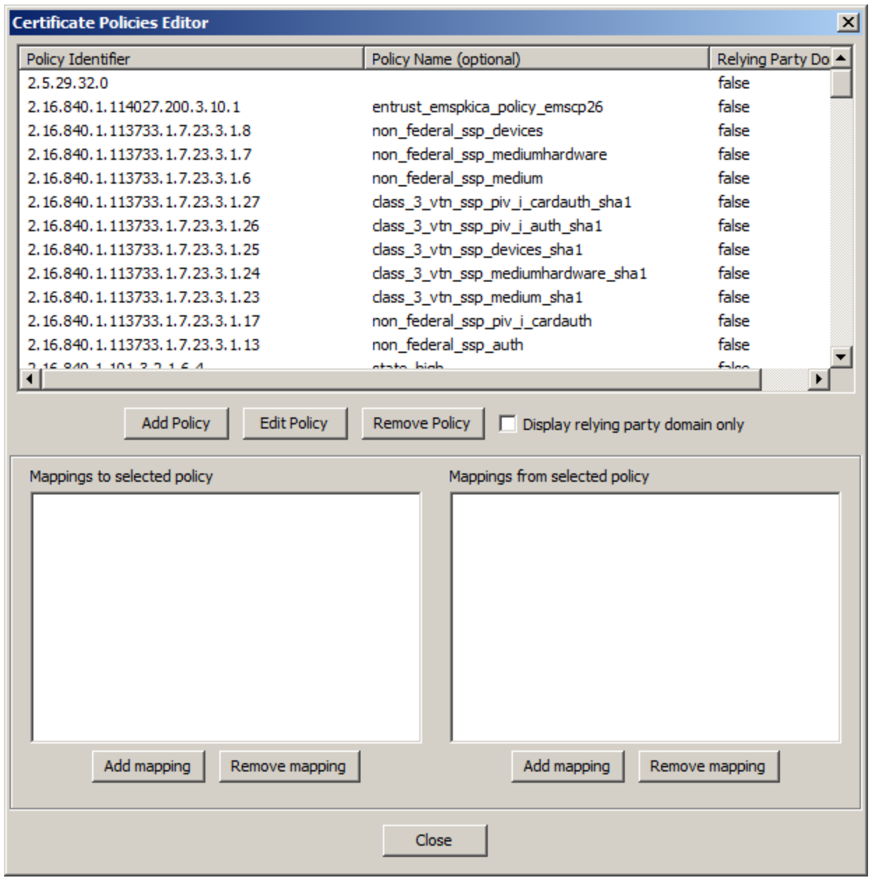
Elliptic curve: 160

DSA: 1024

Figure 29 Defining algorithm prohibitions and minimum key sizes

Edit certificate policies databases

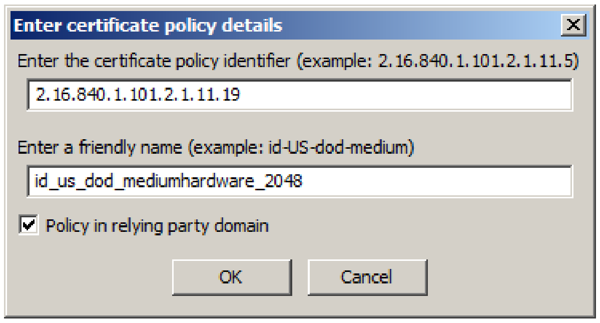
The certificate policies database editor, shown below, can be accessed via the TACT Server Configuration utility options dialog or the TA Store Manager utility options dialog.


Certificate Policies Editor
Add Policy
Edit Policy
Remove Policy
Display relying party domain only
Mappings to selected policy
Mappings from selected policy
Add mapping
Remove mapping
Add mapping
Remove mapping
Close

Policy Identifier	Policy Name (optional)	Relying Party Do
2.5.29.32.0		false
2.16.840.1.114027.200.3.10.1	entrust_emspkica_policy_emscp26	false
2.16.840.1.113733.1.7.23.3.1.8	non_federal_ssp_devices	false
2.16.840.1.113733.1.7.23.3.1.7	non_federal_ssp_mediumhardware	false
2.16.840.1.113733.1.7.23.3.1.6	non_federal_ssp_medium	false
2.16.840.1.113733.1.7.23.3.1.27	class_3_vtn_ssp_piv_i_cardauth_sha1	false
2.16.840.1.113733.1.7.23.3.1.26	class_3_vtn_ssp_piv_i_auth_sha1	false
2.16.840.1.113733.1.7.23.3.1.25	class_3_vtn_ssp_devices_sha1	false
2.16.840.1.113733.1.7.23.3.1.24	class_3_vtn_ssp_mediumhardware_sha1	false
2.16.840.1.113733.1.7.23.3.1.23	class_3_vtn_ssp_medium_sha1	false
2.16.840.1.113733.1.7.23.3.1.17	non_federal_ssp_piv_i_cardauth	false
2.16.840.1.113733.1.7.23.3.1.13	non_federal_ssp_auth	false

Figure 30 Certificate policies database editor

The certificate policies database editor provides a means for editing the contents of a certificate policies database. The schema of the database is provided in Appendix B. To add a new policy, click the **Add Policy** button. To edit a policy that is already present in the database, select the policy in the list then click the **Edit Policy** button. The figure below shows the dialog used when a new certificate policy is added or a certificate policy is edited.


Enter certificate policy details

Enter the certificate policy identifier (example: 2.16.840.1.101.2.1.11.5)

2.16.840.1.101.2.1.11.19

Enter a friendly name (example: id-US-dod-medium)

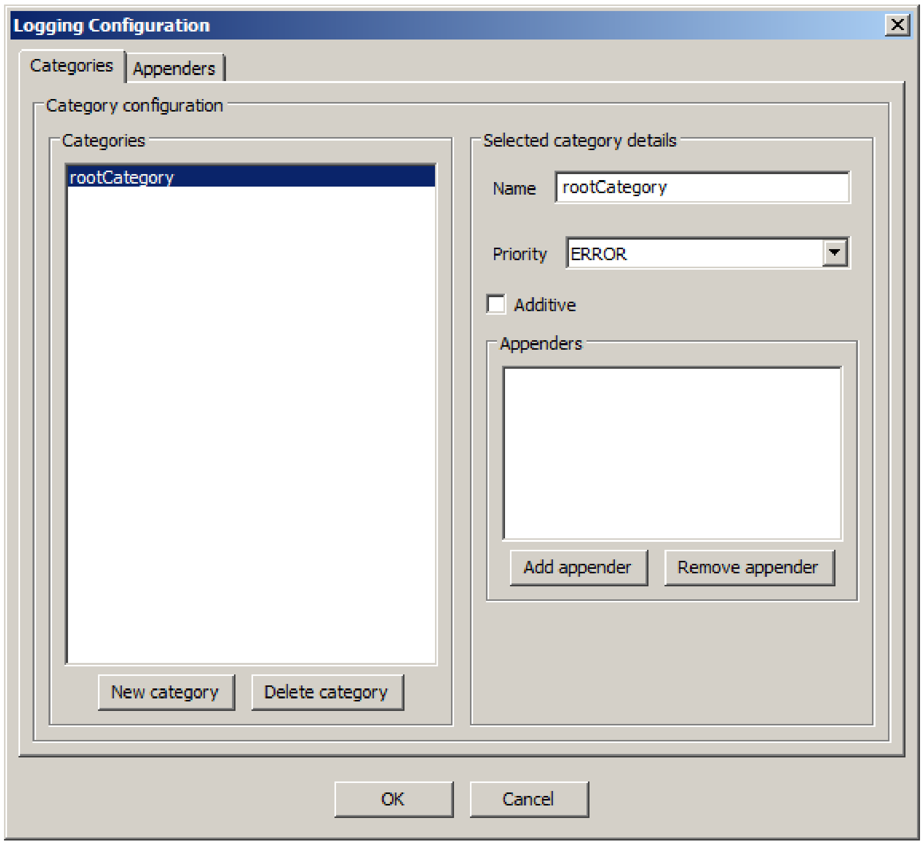
id_us_dod_mediumhardware_2048

☑ Policy in relying party domain

Figure 31 Certificate policy definition

Edit logging configuration

Logging configurations can be edited using the Log Configuration Editor dialog, shown below.


Logging Configuration

rootCategory

Selected category details

Name: rootCategory

Priority: ERROR

☐ Additive

Figure 32 Logging configuration editor

This dialog enables the definition multiple logging categories and appenders. Categories are used by applications to indicate a logging source. Appenders are destinations for log information. Zero or more appenders can be associated with a category. At least one category, rootCategory, must be defined. Additional categories may be defined to direct output from different log sources to different destinations. PITTv2 users may find the following logging categories useful:

- RhUtils
- RhUtilsRev
- PITTv2
- UriProcessing
- PathProcessing

Additional logging categories may also appear but are less likely to be of interest to PITTv2 users:

- RhUtilsAPI
- tact_auth_core
- RhUtilsUi

PITTv2 and TACT use the same logging configuration file format and user interface components to configure logging. The remainder of this section describes the use of the logging configuration user interface relative to TACT's management utilities. The same mechanisms can be used to configure logging for PITTv2.

The basic steps of preparing a logging configuration are as follows:

- Identify the log categories of interest
- Define appenders for log destinations
- Define categories of interest and associated the corresponding appender

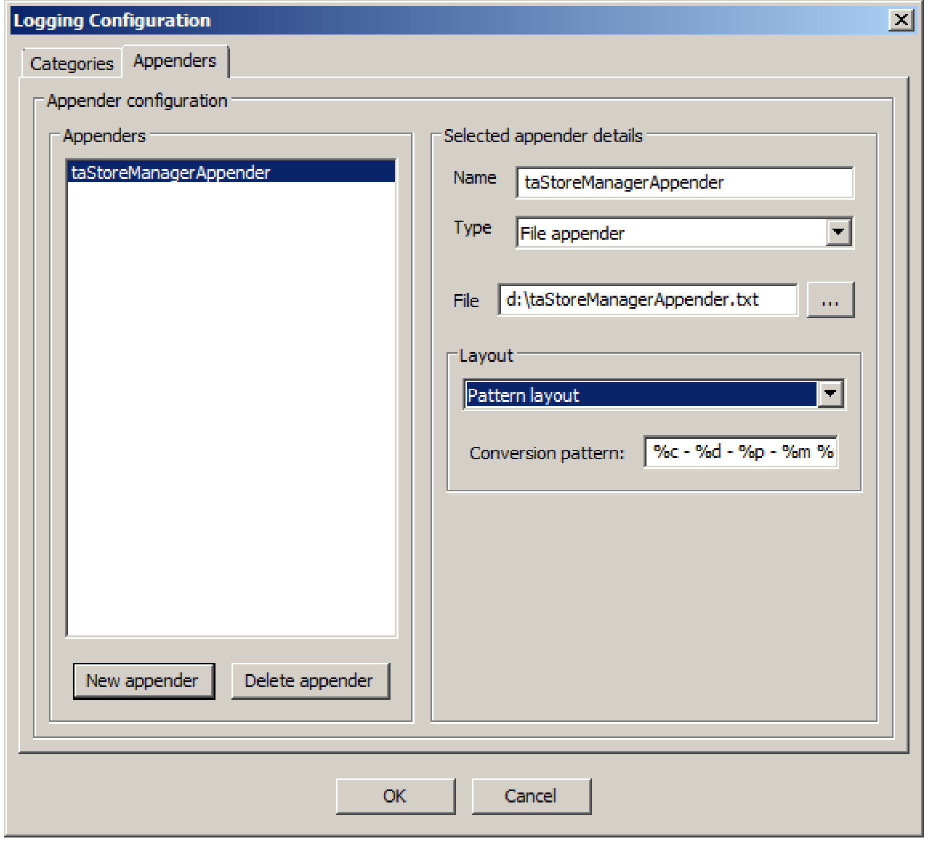
The following steps illustrate how to use the logging configuration editor to prepare a logging configuration file for use by the various management utilities, with each application's output begin written to an independent file, RhUtils output being written to a rolling set of files and TactCli output being written both to a file and the console.

To define an appender to direct log output to a file, perform the following steps:

- Click on the **Appenders** tab
- Click the **New appender** button
- Enter a name for the new appender in the **Name** field

- Select **File appender** from the **Type** drop list
- Enter the full path and filename of the file to receive log information
- Choose a **Layout** (the default Pattern layout is recommended)

The figure below shows the appender details after following the steps above.



Appenders: taStoreManagerAppender

Name: taStoreManagerAppender

Type: File appender

File: d:\taStoreManagerAppender.txt

Layout: Pattern layout

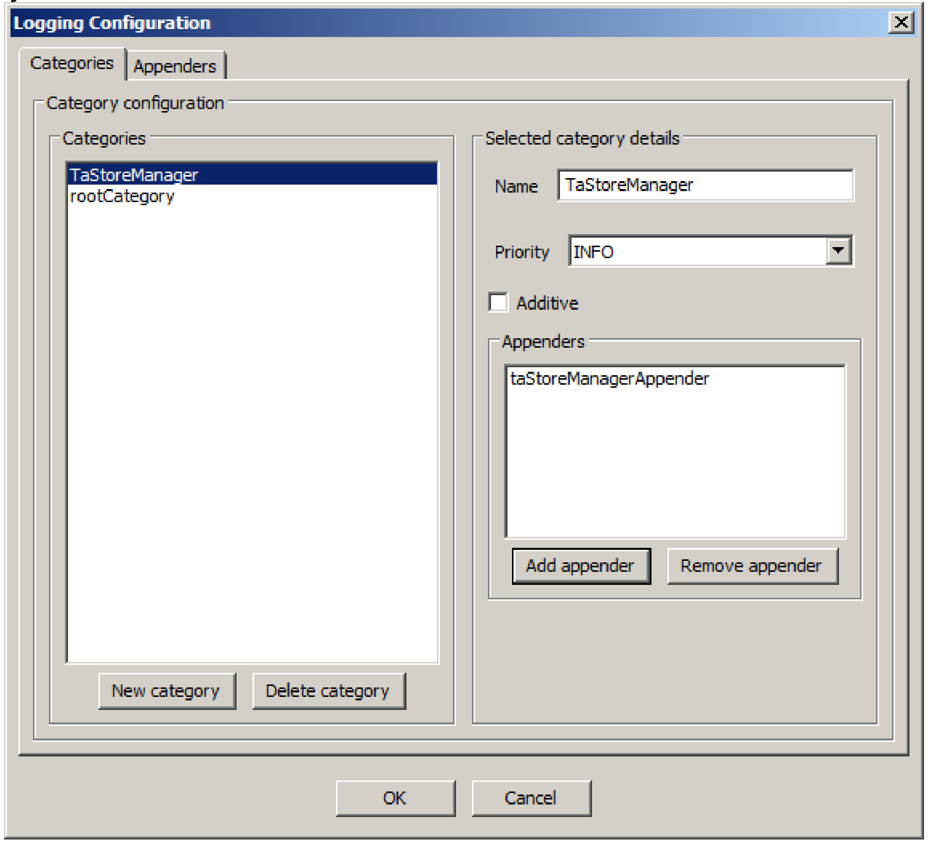
Conversion pattern: %c - %d - %p - %m %

Figure 33 New file appender definition

To associate the new appender with a category to receive TA Store Manager output, perform the following steps:

- Click on the **Categories** tab
- Click the **New category** button
- Enter TaStoreManager in the **Name** field
- Select the desired **Priority** value
- Click the **Add Appender** button
- Select the newly defined appender from the drop list and click **OK**

The figure below shows the category details after following the steps above. In this example, the **Additive** check box is not checked. When this box is checked, the log output is written to each appender associated with the category and to any appenders associated with the rootCategory. In this example, the output is not written to the rootCategory.



Categories: TaStoreManager, rootCategory

Name: TaStoreManager

Priority: INFO

Additive: ☐

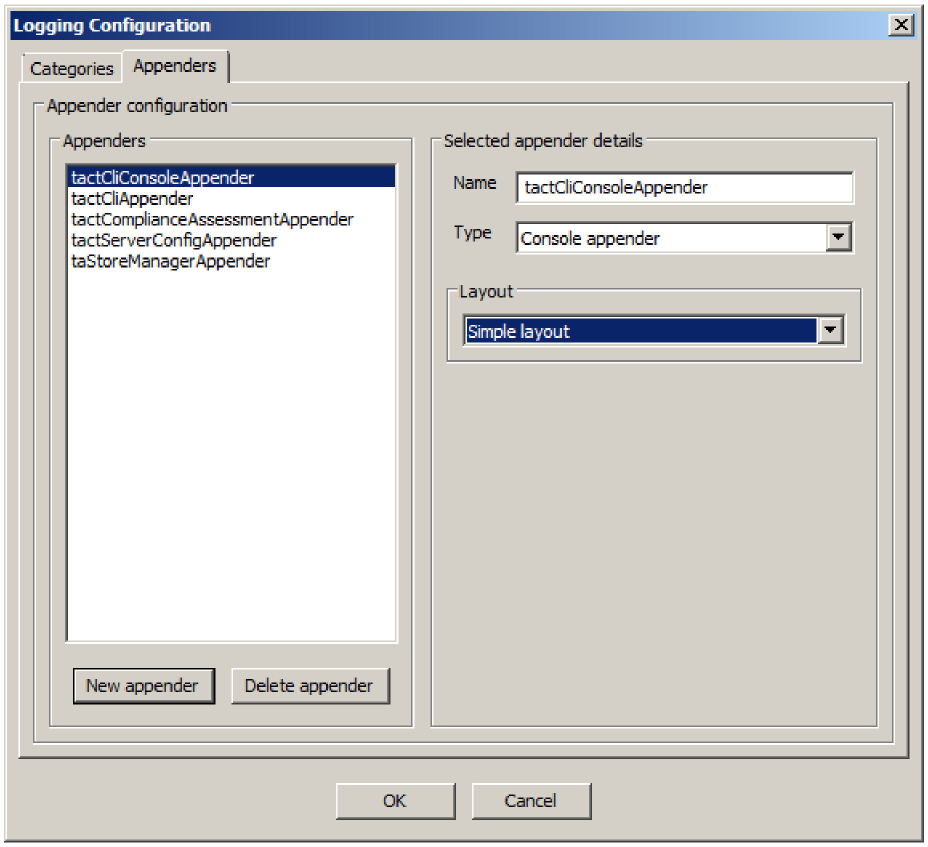
Appenders: taStoreManagerAppender

Figure 34 New category definition

Repeat the steps defined above for creating new file appenders for the TACT Server Configuration utility, TACT Compliance Assessment utility and TACTCli utility. Since TactCli output is to also be written to the console, perform the following steps to prepare a new console appender.

- Click on the **Appenders** tab
- Click the **New appender** button
- Enter a name for the new appender in the **Name** field
- Select **Console appender** from the **Type** drop list
- Choose a **Layout** (Simple layout is recommended for console appenders)

The figure below shows the appender details following execution of the above steps.

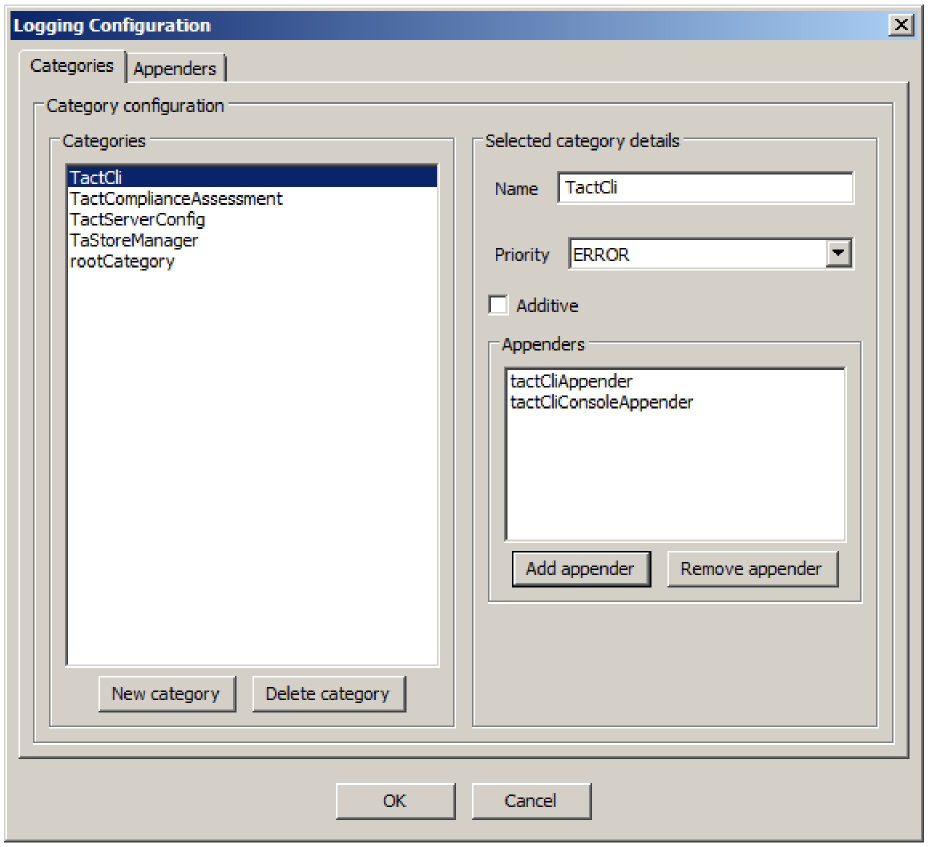


tactCliConsoleAppender
tactCliAppender
tactComplianceAssessmentAppender
tactServerConfigAppender
taStoreManagerAppender

Name: tactCliConsoleAppender
Type: Console appender
Layout: Simple layout

Figure 35 New console appender definition

In the above figure, there is one file appender for each application plus a console appender for TactCli. To associate the new console appender with the TactCli category, follow the steps described above for associating appenders with categories. In this case, the category will feature two appenders, as shown below.



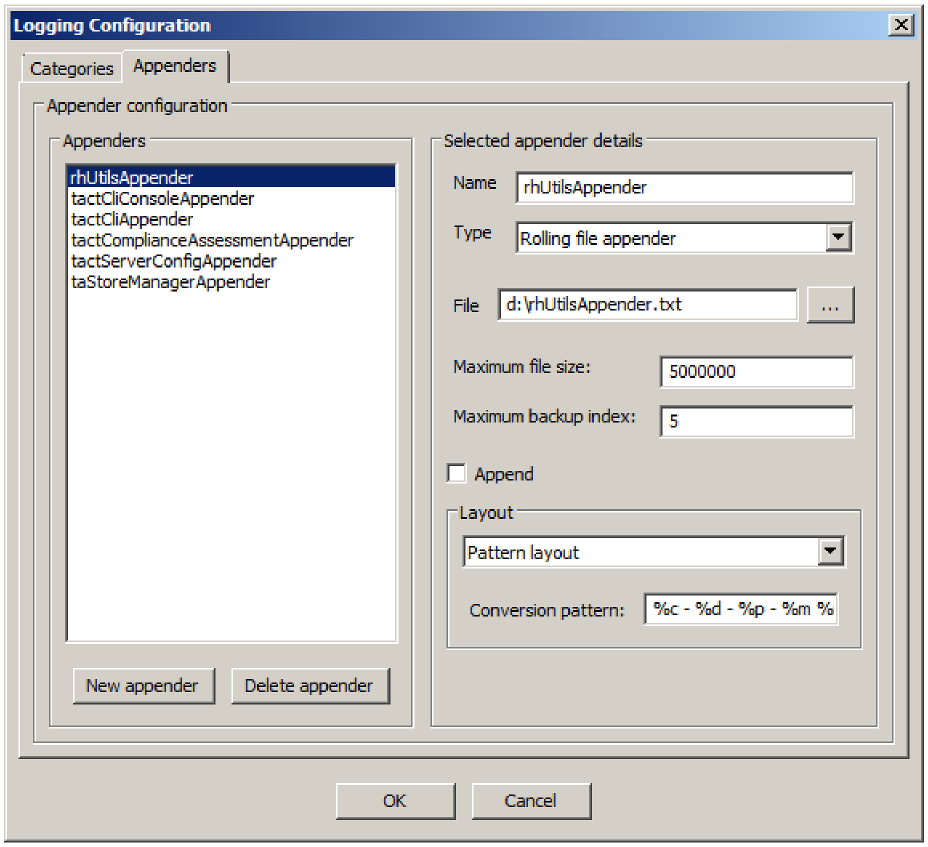
TactCli
TactComplianceAssessment
TactServerConfig
TaStoreManager
rootCategory

Name: TactCli
Priority: ERROR
Additive: [ ]

tactCliAppender
tactCliConsoleAppender

Figure 36 Category definition with multiple appenders

The last steps are to configure RhUtils output to use a rolling file appender and to, optionally, define an appender for the rootCategory. The figure below shows an example appender definition for a rolling file appender with log file size limited to 5MB.



Appender configuration

Appenders:
- rhUtilsAppender
- tactCliConsoleAppender
- tactCliAppender
- tactComplianceAssessmentAppender
- tactServerConfigAppender
- taStoreManagerAppender

Selected appender details:
- Name: rhUtilsAppender
- Type: Rolling file appender
- File: d:\rhUtilsAppender.txt
- Maximum file size: 5000000
- Maximum backup index: 5
- Append: ☐
- Layout: Pattern layout
- Conversion pattern: %c - %d - %p - %m %

Figure 37 New rolling file appender definition

Using the steps above, each management application can share a common logging configuration file with output directed to independent files. Appendix D contains the textual configuration corresponding to the above steps (augmented with a rolling file appender for rootCategory).

An alternative, simpler approach would be to create a logging configuration file per application and define an appender for the rootCategory only, to capture all logging output from the application in one location. The logging configuration approach selected will vary with TACT operator preferences.

In all cases, it is recommended that only the ERROR priority be used for operational situations except for limited troubleshooting periods as DEBUG output can be fairly noisy.

Note, logging configurations may contain platform specific mechanisms, i.e., Event log appender and Win32 debug appender on Windows or Syslog appender on Linux. Logging configurations containing platform specific mechanisms must only be edited or used on the corresponding platform.

Appendix A: Support

Web Site
Please visit the URL below for additional information.
http://iase.disa.mil/pki-pke

Technical Support
Contact technical support at the email address below.
dodpke@mail.mil

Appendix B: File Formats

PITTv2 uses the same file formats used by TACT v1.2, except for TAMP messages which may be used by TA Store Manager or TactCli to manage a trust anchor store used by PITTv2. TACT uses a number of different types of files. The table below provides a summary overview of these different file types. The following sections provide additional detail for selected file types.

File type	File extensions	Description
TACT Settings	*.sdb	Contains path settings, security environment and protected resource definitions. Primarily edited using the TACT Server Configuration utility.
TACT Trust Anchor Store	*.tas	Contains trust anchors. Edited using the TA Store Manager utility and TactCli utility (when using TAMP messages).
Certificate Policies	*.pdb	Contains certificates policies, certificate policy descriptions and policy mappings. Primarily edited using the TA Store Manager utility or TACT Server Configuration utility.
Path Settings	*.ps	Contains path settings definition
Security Environment	*.se	Contains security environment definition
TACT Resource	*.tr	Contains protected resource definition
Logging Configuration	*.conf; *.properties	Contains logging configuration
PKI Log	*.db	Contains certificates, certification paths and other session-related information
URI last modified	uriLastModified.db	Automatically generated file that maintains the last modified time for resources obtained from an indicated URI
CRL file info	crlIndex.db	Automatically generated file that maps CRL information to a file name
Trust Anchors	*.ta	Contains an RFC 5914

		TrustAnchorChoice object
X.509 Certificates	*.der; *.crt; *.cer	Contains an RFC 5280 Certificate object
TAMP Update	*.tur; *.tampUpdate	Contains an RFC 5934 TAMPUpdate message (wrapped in a ContentInfo or SignedData)
TAMP Status Response	*.tsr; *.statusResponse	Contains an RFC 5934 TAMPStatusResponse message (wrapped in a ContentInfo or SignedData)

TACT utilities place limit the size of each file type. Prior to opening a file, the size is checked and if the limit is exceeded an error message is displayed or logged. The following limits apply, each of which is significantly larger than expected operational sizes:

- TACT Settings: 25MB
- TACT Trust Anchor Store: 10MB
- Certificate Policies: 10MB
- Path Settings: 2MB
- Security Environment: 2MB
- Logging Configuration: 2MB
- Trust Anchors: 2MB
- X.509 Certificates: 2MB
- TAMP Update: 10MB
- TAMP Status Response: 10MB

TACT Settings

TACT settings databases contain all configuration information for a TACT plugin installation, including certification path validation settings, basic security environment settings and protected resource definitions. TACT settings databases are SQLite databases that conform to the schema shown below. This schema is provided for informational purposes only. TACT settings databases should only be edited using the provided utilities.

```
CREATE TABLE PathSettings (
    PathSettingsId INTEGER PRIMARY KEY AUTOINCREMENT,
    PathSettingsName TEXT UNIQUE,
    PathSettings BLOB,
    LastChange DATETIME);
CREATE TABLE SecurityEnvironments (
    SecurityEnvironmentId INTEGER PRIMARY KEY AUTOINCREMENT,
    SecurityEnvironmentName TEXT UNIQUE,
    SecurityEnvironment BLOB,
    LastChange DATETIME);
CREATE TABLE Resources (
    ResourceId INTEGER PRIMARY KEY AUTOINCREMENT,
    ResourceName TEXT UNIQUE,
    PathSettings INTEGER OPTIONAL,
    SecurityEnvironment INTEGER OPTIONAL,
    LastChange DATETIME,
    FOREIGN KEY(PathSettings) REFERENCES PathSettings(PathSettingsId) ON DELETE SET NULL,
    FOREIGN KEY(SecurityEnvironment) REFERENCES
SecurityEnvironments(SecurityEnvironmentId) ON DELETE SET NULL);
PRAGMA user_version=1;
PRAGMA foreign_keys = ON;
CREATE TRIGGER [delete_PathSettings]
    BEFORE DELETE ON [PathSettings]
    FOR EACH ROW
    BEGIN
    UPDATE Resources SET PathSettings = NULL WHERE Resources.PathSettings =
old.PathSettingsId;
END;
CREATE TRIGGER [delete_SecurityEnvironments]
    BEFORE DELETE ON [SecurityEnvironments]
    FOR EACH ROW
    BEGIN
    UPDATE Resources SET SecurityEnvironment = NULL WHERE Resources.SecurityEnvironment =
old.SecurityEnvironmentId;
END;
```

TACT Trust Anchor Store

TACT trust anchor stores contain trust anchors for use by TACT plugins. TACT trust anchor stores are SQLite databases that conform to the schema shown below. TACT v1.0 uses only the TrustAnchors and SequenceNumbers tables. This schema is provided for informational purposes only. TACT trust anchor stores should only be edited using the Trust Anchor Store Manager utility.

```
CREATE TABLE TrustAnchors (
       TrustAnchorId INTEGER PRIMARY KEY AUTOINCREMENT,
       SubjectPublicKeyInfo BLOB,
       TrustAnchor BLOB,
       Name TEXT,
       SKID TEXT UNIQUE);
CREATE TABLE SequenceNumbers (
       SequenceNumberId INTEGER PRIMARY KEY AUTOINCREMENT,
       SKID TEXT UNIQUE,
       SequenceNumber INTEGER);
CREATE TABLE TaStoreAdminCertificates (
       CertificateId INTEGER PRIMARY KEY AUTOINCREMENT,
       Certificate BLOB);
CREATE TABLE StoreInformation (
       TrustAnchorStoreName TEXT,
       ApexTrustAnchor BLOB OPTIONAL,
       MaxNumberOfTrustAnchors INTEGER OPTIONAL);
```

```
PRAGMA user_version=1;
```

Certificate Policies

The certificate policies database can be used to simplify configuration of certificate policy related constraints by providing textual descriptions in addition to object identifiers. Additionally, configuration can be performed in terms of certificate policies from the enterprise of the administrator. The policies database is a SQLite database that conforms to the schema shown below. In some environments, creation of a certificate policies database may be performed using custom SQLite scripts or tools to import existing policy information into a new database instance.

```
CREATE TABLE CertificatePolicies (
    CertPolicyId INTEGER PRIMARY KEY AUTOINCREMENT,
    PolicyOid TEXT UNIQUE,
    InRelyingPartyDomain BOOL,
    PolicyName TEXT OPTIONAL);
CREATE TABLE PolicyMappings (
    IssuerDomain INTEGER,
    SubjectDomain INTEGER,
    PRIMARY KEY (IssuerDomain, SubjectDomain),
    FOREIGN KEY(IssuerDomain) REFERENCES CertificatePolicies(CertPolicyId) ON DELETE
CASCADE,
    FOREIGN KEY(SubjectDomain) REFERENCES CertificatePolicies(CertPolicyId) ON DELETE
CASCADE);
PRAGMA user_version=1;
PRAGMA foreign_keys = ON;
CREATE TRIGGER [delete_CertificatePolicy]
    BEFORE DELETE ON [CertificatePolicies]
    FOR EACH ROW
    BEGIN
    DELETE FROM PolicyMappings WHERE PolicyMappings.IssuerDomain = old.CertPolicyId;
    DELETE FROM PolicyMappings WHERE PolicyMappings.SubjectDomain = old.CertPolicyId;
    END;
```

Path Settings

Path settings configuration objects contain various settings that influence certification path validation. Path settings configuration files contain XML generated and processed by Boost serialization components. These files should not be edited manually.

Security Environment

Security environment configuration objects contain various settings that influence certification path validation. Security environment values are generally more broadly applicable than those contained in path settings configuration objects. Security environment configuration files contain XML generated and processed by Boost serialization components. These files should not be edited manually.

TACT Resource

TACT resource configuration objects contain various settings that influence access control decisions. TACT resource configuration files contain XML generated and processed by Boost serialization components. These files should not be edited manually.

Logging Configuration

TACT uses the log4cpp library for logging. The log4cpp library uses a configuration file similar to the file used by the log4j library. While these files are typically hand edited, TACT provides a graphical user interface for editing these. It is strongly recommended that these files be edited using the provided tools. In some cases, especially when moving logging configuration files from one platform to another, it may be necessary to edit the files using a text editor to remove platform specific mechanisms.

PKI Log

The PKI log is automatically populated by the TACT plugin during operation. The PKI log is a SQLite database that conforms to the schema shown below. The schema is provided to enable the creation of scripts and other analysis tools for use in reviewing PKI log contents. No analysis tool is provided with TACT.

```
CREATE TABLE Certificates (
       CertificateId INTEGER PRIMARY KEY AUTOINCREMENT,
       Certificate BLOB,
       IssuerName TEXT,
       NotBefore TEXT,
       NotAfter TEXT,
       SubjectName TEXT,
       CertHash TEXT NOT NULL UNIQUE,
       isCA INTEGER,
       SKID TEXT,
       AKID TEXT OPTIONAL,
       certificatePolicies TEXT OPTIONAL,
       certificateType TEXT OPTIONAL,
       edipi TEXT OPTIONAL
);
CREATE TABLE AssociatedPaths (
       PartialCertificationPath NUMERIC,
       Certificate NUMERIC,
       PRIMARY KEY (PartialCertificationPath, Certificate),
       FOREIGN KEY(PartialCertificationPath) REFERENCES
PartialCertificationPaths(PartialCertificationPathId),
       FOREIGN KEY(Certificate) REFERENCES Certificates(CertificateId)
);
CREATE TABLE PartialCertificationPaths (
       PartialCertificationPathId INTEGER PRIMARY KEY AUTOINCREMENT,
       PathHash TEXT,
       NumCerts NUMERIC
);
CREATE TABLE PartialCertificationPathMembers (
       PartialCertificationPath NUMERIC,
       Certificate NUMERIC,
       PathIndex NUMERIC,
       PRIMARY KEY (PartialCertificationPath, Certificate, PathIndex),
       FOREIGN KEY(PartialCertificationPath) REFERENCES
PartialCertificationPaths(PartialCertificationPathId),
       FOREIGN KEY(Certificate) REFERENCES Certificates(CertificateId)
);
CREATE TABLE AccessLog (
       EventId INTEGER PRIMARY KEY AUTOINCREMENT,
       Time TEXT,
       SourceIP TEXT,
       SourcePort INTEGER,
       DestIP TEXT,
       DestPort INTEGER,
       UrlAccessed TEXT,
       AccessAllowed INTEGER,
       CertId NUMERIC,
       PathId NUMERIC OPTIONAL,
    FOREIGN KEY(CertId) REFERENCES Certificates(CertificateId),
    FOREIGN KEY(PathId) REFERENCES PartialCertificationPaths(PartialCertificationPathId)
);
PRAGMA user_version=1;
```

URI Last Modified

The UriLastModified database is populated when a resource is successfully downloaded from a server via HTTP. If the server provides a last modified time for the requested resource, the value is stored in the database. Subsequent requests for the same resource will include the stored time to avoid downloading resources that have not changed since previously obtained (and notionally still locally available). The uriLastModified.db files can be deleted with no penalty other than fresh retrieval of all remote artifacts associated with the database instance. The file should be deleted (or selectively edited) when automatically downloaded artifacts are manually deleted, else the artifacts may not be downloaded when needed due to a cached last modified time.

```
CREATE TABLE UriLastModified (
       UriId INTEGER PRIMARY KEY AUTOINCREMENT,
       Uri TEXT UNIQUE,
       LastModified TEXT);
```

CRL Info

The crlIndex.db file is created to index CRLs present within a given folder. The index maps information useful in locating a CRL appropriate for determining the revocation status of a given certificate to a file name.

```
CREATE TABLE CrlFileInfo (
    CrlFileId INTEGER PRIMARY KEY AUTOINCREMENT,
    FullPathAndFilename TEXT,
    SKID TEXT,
    ThisUpdate BLOB,
    NextUpdate BLOB OPTIONAL,
    IssuerName TEXT,
    IssuerNameBlob BLOB,
    SigAlgBlob BLOB,
    CrlScope INTEGER,
    CrlCoverage INTEGER,
    CrlAuthority INTEGER,
    CrlReasons INTEGER,
    Extensions BLOB OPTIONAL,
    DistPointName TEXT OPTIONAL,
    DistPointNameBlob BLOB OPTIONAL,
    DeltaSeqNumber TEXT OPTIONAL
);
```

Appendix C: Error Codes

The certificate validation dialog in the TACT Server Configuration Utility displays error codes reported by the certification path validation library. A description of each error code is below.

Error code name	Value	Description
NAME_CHAINING_FAILURE	25001	The subject name in a CA certificate or trust anchor does not match the issuer name in an immediately subordinate certificate in a certification path.
SIGNATURE_VERIFICATION_FAILURE	25002	The signature on a certificate could not be verified using the public key information from the immediately superior certificate in a certification path.
INVALID_NOT_BEFORE_DATE	25003	The not before value in a certificate in a certification path is later than the validation time of interest.
INVALID_NOT_AFTER_DATE	25004	The not after value in a certificate in a certification path is earlier than the validation time of interest.
MISSING_BASIC_CONSTRAINTS	25005	A purported CA certificate in a certification path is missing the basic constraints extension.
INVALID_BASIC_CONSTRAINTS	25006	A purported CA certificate in a certification path contains an invalid basic constraints extension value.
INVALID_PATH_LENGTH	25007	A certification path violated a path length constraint.
INVALID_KEY_USAGE	25008	A certificate contains an invalid key usage. For example, a CA certificate may not feature a key usage value that permits certificate signing.
NULL_POLICY_SET	25009	The valid policy tree became NULL during certification path validation.
NAME_CONSTRAINTS_VIOLATION	25010	A certificate in a certification path contains a name that violates a name constraint.
UNPROCESSED_CRITICAL_EXTENSION	25011	A certificate in a certification path contains a critical extension that was not processed.
CCC_UNAUTH_TA	25012	A CMS content constraints error was detected.
CCC_VIOLATION	25013	A CMS content constraints error was detected.
MISSING_TRUST_ANCHOR	25014	Certification path validation was attempted for a certification path that does not include a trust anchor.
MISSING_TRUST_ANCHOR_NAME	25015	A certification path includes a trust anchor that does not include a name and is thus not suitable for validating certification paths.
PROHIBITED_ALG	25016	A certificate in a certification path contains a public key or signature value associated with a prohibited algorithm.
PROHIBITED_KEY_SIZE	25017	A certificate in a certification path contains a public key that violates a key size constraint.
ENCODING_ERROR	25018	A BER/DER error was detected.
MISSING_CERTIFICATE	25019	A CMS SignedData message does not include a certificate required to validate the signature.
UNEXPECTED_CONTENT_TYPE	25020	A CMS message contains an unrecognized content type.
SEQ_NUM_VIOLATION	25021	A TAMP message contains a sequence number that is lower than the sequence number value stored for the message signer, i.e., the message is stale.
NO_PATHS_FOUND	25022	A certification path could not be found from a given end entity certificate to a trust anchor.
COUNTRY_CODE_VIOLATION	25023	The certification path includes a certificate that violates a country code constraint expressed in the indicated PathSettings.
CERTIFICATE_REVOKED	25024	The certification path includes a certificate that has been revoked.
REVOCATION_STATUS_NOT_DETERMINED	25025	The certification path contains one or more certificates for which revocation status could not be determined.
CERTIFICATE_ON_HOLD	25026	The certification path contains a certificate that has been placed on hold.
CERTIFICATE_BLACKLISTED	25027	The certification path includes a certificate that has been blacklisted in the specified revocation configuration.
STATUS_CHECK_RELIED_ON_STALE_CRL	25028	Revocation status checks utilized at least one stale CRL.
REVOCATION_STATUS_NOT_AVAILABLE	25029	No revocation status information was found for the certificate being

		checked.

Appendix D: Sample Log Configuration

The following logging configuration corresponds to the example in the Edit logging configuration section above.

```
log4j.appender.RollingPath=org.apache.log4j.RollingFileAppender
log4j.appender.RollingPath.append=true
log4j.appender.RollingPath.fileName=e:\PITT_testing\Logs\Windows\PITTPath.txt
log4j.appender.RollingPath.layout=org.apache.log4j.PatternLayout
log4j.appender.RollingPath.layout.ConversionPattern=%c - %d - %p (%t) - %m %n
log4j.appender.RollingPath.maxBackupIndex=10
log4j.appender.RollingPath.maxFileSize=1000000
log4j.appender.RollingPitt=org.apache.log4j.RollingFileAppender
log4j.appender.RollingPitt.append=true
log4j.appender.RollingPitt.fileName=E:\PITT_testing\Logs\Windows\Pittv2.txt
log4j.appender.RollingPitt.layout=org.apache.log4j.PatternLayout
log4j.appender.RollingPitt.layout.ConversionPattern=%c - %d - %p (%t) - %m %n
log4j.appender.RollingPitt.maxBackupIndex=10
log4j.appender.RollingPitt.maxFileSize=5000000
log4j.appender.RollingPittAll=org.apache.log4j.RollingFileAppender
log4j.appender.RollingPittAll.append=true
log4j.appender.RollingPittAll.fileName=e:\PITT_testing\Logs\Windows\PittAll.txt
log4j.appender.RollingPittAll.layout=org.apache.log4j.PatternLayout
log4j.appender.RollingPittAll.layout.ConversionPattern=%c - %d - %p (%t) - %m %n
log4j.appender.RollingPittAll.maxBackupIndex=10
log4j.appender.RollingPittAll.maxFileSize=5000000
log4j.appender.RollingUri=org.apache.log4j.RollingFileAppender
log4j.appender.RollingUri.append=true
log4j.appender.RollingUri.fileName=e:\PITT_testing\Logs\Windows\PittUri.txt
log4j.appender.RollingUri.layout=org.apache.log4j.PatternLayout
log4j.appender.RollingUri.layout.ConversionPattern=%c - %d - %p (%t) - %m %n
log4j.appender.RollingUri.maxBackupIndex=10
log4j.appender.RollingUri.maxFileSize=1000000
log4j.appender.RollingUtils=org.apache.log4j.RollingFileAppender
log4j.appender.RollingUtils.append=true
log4j.appender.RollingUtils.fileName=E:\PITT_testing\Logs\Windows\PittUtils.txt
log4j.appender.RollingUtils.layout=org.apache.log4j.PatternLayout
log4j.appender.RollingUtils.layout.ConversionPattern=%c - %d - %p (%t) - %m %n
log4j.appender.RollingUtils.maxBackupIndex=10
log4j.appender.RollingUtils.maxFileSize=5000000
log4j.category.PITTv2=INFO,RollingPitt
log4j.category.PathProcessing=DEBUG,RollingPath
log4j.category.RhUtils=DEBUG,RollingUtils
log4j.category.RhUtilsAPI=ERROR
log4j.category.UriProcessing=DEBUG,RollingUri
log4j.category.tact_auth_core=ERROR
log4j.rootCategory=DEBUG,RollingPittAll
```

Appendix E: Definitions

Path settings	Configuration item that includes: • RFC 5280 certification path validation input definitions • RFC 5937 certification path validation input definitions • Cryptographic algorithm prohibition and cryptographic key size constraints enforcement indicator
Security environment	Configuration item that defines: • TACT trust anchor store to use during certification path validation • Certificate policies database to use during path settings configuration and logging of certification path validation results • Cryptographic algorithm prohibitions • Cryptographic key size constraints
TACT	Trust Anchor Constraints Tool
TACT administrator	Prepares TACT settings for use in defining TACT resources using the TACT Server Configuration Utility
TACT operator	Web server administrator who uses a TACT settings database with path settings and security environment definitions provided by a TACT administrator to define TACT resources in the TACT Server Configuration Utility
TACT resource	A server resource subject to a path settings configuration and a security environment configuration. Expressed as a URI path or a file system path, for example: • / • C:\inetpub\wwwroot • /var/www/html
TACT settings	A collection of path settings and security environment configurations that may be used to define TACT resources
TASM	TA Store Manager utility
TA store administrator	Prepares and maintains trust anchor stores using the TA Store Manager utility. Uses signed TAMP messages for remote management.
TCA	TACT Compliance Assessment utility
Trust anchor	An RFC 5914 TrustAnchorChoice containing a TrustAnchorInfo structure that wraps an X.509 certificate

TSC	TACT Server Configuration utility

Appendix F: Bibliography

RFC 5280	Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.
RFC 5652	Housley, R., "Cryptographic Message Syntax (CMS)", RFC 5652, September 2009.
RFC 5914	Housley, R., Ashmore, S., and C. Wallace, "Trust Anchor Format", RFC 5914, June 2010.
RFC 5934	Housley, R., Ashmore, S., and C. Wallace, "Trust Anchor Management Protocol (TAMP)", RFC 5934, August 2010.
RFC 5937	Ashmore, S., and C. Wallace, "Using Trust Anchor Constraints during Certification Path Processing", RFC 5937, August 2010.

Appendix G: Sample Scripts

Importing and exporting certificate policy information

The following sample Python script enables certificate policy information to be defined using a Comma-Separated Values (CSV) file and imported into a TACT certificate policies database or for information in a TACT certificate policies database to be exported to a CSV file.

```
import sqlite3
import csv
import logging
from optparse import OptionParser,OptionError

# getPolicyOid takes an object identifier in dot notation form (i.e., 1.2.3.4.5) and returns
# the integer ID of the OID from the database, if any.  If the OID is not found, -1 is returned.
def getPolicyId(conn, oid):
        c = conn.cursor()
        query = "select CertPolicyId from CertificatePolicies WHERE PolicyOid='" + oid + "';"
        c.execute(query)
        id = -1
        for row in c:
           id = row[0]
        c.close()
        return id

# getPolicyOid takes the ID of a policy in the database and returns the object identifier.
def getPolicyOid(conn, id):
        c = conn.cursor()
        query = "select PolicyOid from CertificatePolicies WHERE CertPolicyId=" + str(id) + ";"
        c.execute(query)
        oid = ""
        for row in c:
           oid = row[0]
        c.close()
        return oid

# addPolicy takes an object identifier, a indication of association with the relying party domain
# and a descriptive name and adds a new item to the database.
def addPolicy(conn, oid, irpd, name):
        c = conn.cursor()
        query = "INSERT INTO CertificatePolicies VALUES(NULL,'" + oid + "'," + irpd + ",'" + name + "');"
        print query
        c.execute(query)
        conn.commit()
        c.close()

# addMapping takes a pair of database identifiers that identify policies present in the
# CertificatePolicies tables and creates a new entry in the PolicyMappings table where id
# is the IssuerDomain and mappedId is the subject domain,
def addMapping(conn, id, mappedId):
        print "addMapping " + str(id) + " " + str(mappedId)
        c = conn.cursor()
        query = "INSERT INTO PolicyMappings VALUES(" + str(id) + "," + str(mappedId) + ");"
        print query
        c.execute(query)
        conn.commit()
        c.close()

# getMappings takes an object identifier in dot notation form (i.e., 1.2.3.4.5) and returns
# a string containing a semi-colon delimited list of object identifiers the provided OID
# maps to.
```

```
def getMappings(conn, oid):
        mappings = ""
        id = getPolicyId(conn, oid)
        if(-1 == id):
           return mappings;

        first = True
        c = conn.cursor()
        c.execute("select SubjectDomain from PolicyMappings WHERE IssuerDomain=" + str(id) + ";")
        for row in c:
           oid = getPolicyOid(conn, row[0])
           if(first):
                        mappings = oid
                        first = False
           else:
                        mappings += ";" + oid
        c.close()
        return mappings

# addMappings takes an object identifier and a string containing a semi-colon delimited list
# of object identifiers.  It creates an entry in the PolicyMappings table for each element
# in the delimited list where oid is the IssuerDomain and each element is a SubjectDomain
# policy.
def addMappings(conn, oid, mappings):
        print "addMappings " + oid + " " + mappings
        id = getPolicyId(conn, oid)
        if(-1 == id):
           return
        for item in mappings.split(";"):
           mappedId = getPolicyId(conn, item)
           if(-1 != mappedId):
                        addMapping(conn,id, mappedId)

# exportToCsv exports policies from the database identified by the --database option and writes
# the information in comma-delimited form to the file identified by the --csv option.
def exportToCsv(options):
        print "exportToCsv"
        writer = csv.writer(open(options.csv,'wb+'), delimiter=',')
        conn = sqlite3.connect(options.database)
        c = conn.cursor()
        c.execute("select PolicyOid, InRelyingPartyDomain, PolicyName from CertificatePolicies;")
        for row in c:
           mappings = getMappings(conn, row[0])
           writer.writerow([row[0], row[1], row[2],mappings])
        c.close()

# importFromCsv imports policies into the database identified by the --database option from a
# comma-delimited CSV file identified by the --csv option.
def importFromCsv(options):
        print "importFromCsv"
        conn = sqlite3.connect(options.database)
        for row in reader:
           oid = row[0]
           irpd = row[1]
           name = row[2]
           addPolicy(conn, oid, irpd, name)
        reader = csv.reader(open(options.csv,'rU'), delimiter=',')
        for row in reader:
           oid = row[0]
           mappings = row[3]
           addMappings(conn, oid, mappings)

def main():
    logger = logging.getLogger('')
    logger.setLevel(logging.INFO)
    handler = logging.StreamHandler()
    handler.setLevel(logging.DEBUG)
    formatter = logging.Formatter('%(asctime)s [%(levelname)s] %(message)s')
    handler.setFormatter(formatter)
    logger.addHandler(handler)

    usage = "Usage: %prog [options]"
    parser = OptionParser(usage)
    parser.add_option("-e","--exporting",dest="exporting",
            help="when true data is exported from database to CSV, vice versa when false")
    parser.add_option("-d","--database",dest="database",
            help="file containing certificate policies database")
    parser.add_option("-c","--csv",dest="csv",
            help="file containing CSV formatted certificate policy information")
    try:
        (options,args) = parser.parse_args()
    except OptionError,e:
        logger.error("Unable to parse command line options: %s" % e.msg)
        sys.exit(1)

    print options
    if(options.exporting == "True"):
            exportToCsv(options)
    else:
            importFromCsv(options)

if __name__ == "__main__":
    main()
```

Downloading and applying a TAMP update message

The following Windows PowerShell script demonstrates how to download a TAMP update message for processing with the TactCli utility.

```
function main()
{
                $clnt = new-object System.Net.WebClient
                $urlbase = "http://locker.redhoundsoftware.com:8080/"
                $filename = "tu.tur"
                $targetdir = "C:\TactSettings\"

                $url = $urlbase + $filename
                $outfile = $targetdir + $filename
                $stamp =  ( Get-Date ( Get-Date ).AddDays(-1) -uformat %m%d%Y%H%M%S )
                if(Test-Path $outfile)
                {
                                write-host 'moving existing ' $filename ' out of the way.'
                                $backupfile = $targetdir + $filename + "." + $stamp + ".bak"
                                rename-item $outfile $backupfile
                }
                write-host 'downloading ' $url
                $clnt.DownloadFile($url,$outfile)

                $TactCli="C:\Program Files\TACT\bin\TactCli.exe"
                $database = $targetdir + "tas\default.tas"
                $logconfig = $targetdir + "tclog.properties"
                & "$TactCli" --action update --tampUpdate $outfile --database $database -l $logconfig
}

main
exit 0
```

Retrieving information from the PKI log

The text logs generated by the TACT plugins include the event ID associated with a particular access event. Additional information can be retrieved from the database given this event ID. The sample Python script below can be used to retrieve access information, optionally including the certification path.

```
import sys
import sqlite3
import logging
from argparse import ArgumentParser,ArgumentError

def DumpCert(certId, index, results):
        conn = sqlite3.connect(results.database)
        c = conn.cursor()
        query = "select Certificate, SubjectName from Certificates WHERE CertificateId = " + str(certId) + ";"
        c.execute(query)
        if 0 == c.rowcount:
           c.close()
           return

        for row in c:
           fileName = results.outputFolder + "Event" + str(results.eventID) + "-Cert" + str(index) + ".der"
           f = open(fileName, 'wb')
           f.write(row[0])
           print "* Certificate #" + str(index) + " - " + row[1]
        c.close()

def DumpCerts(pathId, certId, results):
        conn = sqlite3.connect(results.database)
        c = conn.cursor()
        query = "select Certificate,PathIndex from PartialCertificationPathMembers WHERE PartialCertificationPath = " + str(pathId)
+ " ORDER BY PathIndex DESC;"
        c.execute(query)
        print "Certificate information for event #" + str(results.eventID)
        index = 0
        for row in c:
           DumpCert(row[0], index, results)
           index = index + 1
        c.close()
        DumpCert(certId, index, results)

def main():
   logger = logging.getLogger('')
   logger.setLevel(logging.INFO)
   handler = logging.StreamHandler()
   handler.setLevel(logging.DEBUG)
   formatter = logging.Formatter('%(asctime)s [%(levelname)s] %(message)s')
   handler.setFormatter(formatter)
   logger.addHandler(handler)

   usage = "Usage: %prog -d -e [-o] [-i] [-h]"
   parser = ArgumentParser(usage)
   parser.add_argument("-e","--eventID",action='store',dest="eventID", default=-1, help="identifies the event to retrieve")
   parser.add_argument("-d","--database",action='store',dest="database", help="file containing PKI log database",type=str)
   parser.add_argument("-o","--outputFolder",action='store',dest="outputFolder", help="folder to receive event information")
   parser.add_argument("-i","--infoOnly",action="store_true",dest="infoOnly",default=False, help="retrieve event information only -
do not return certificates")
   try:
      results = parser.parse_args()
   except ArgumentError,e:
      logger.error("Unable to parse command line options: %s" % e.msg)
      sys.exit(1)
```

```
    if -1 == results.eventID:
        logger.error("No event ID was specified.")
        parser.print_help()
        sys.exit(1)

    if None == results.database:
        logger.error("No database was specified.")
        parser.print_help()
        sys.exit(1)

    if False == results.infoOnly and None == results.outputFolder:
        logger.error("No outputFolder was specified.")
        parser.print_help()
        sys.exit(1)

    conn = sqlite3.connect(results.database)
    c = conn.cursor()
    query = "select * from AccessLog WHERE EventId = " + str(results.eventID) + ";"
    c.execute(query)
    pathId = -1
    certId = -1

    if 0 == c.rowcount:
        print "No event found with ID " + str(results.eventID)
        sys.exit(1)

    print ""
    print "Access information for event #" + str(results.eventID)
    for row in c:
        print "* Time:\t\t\t" + row[1]
        print "* SourceIP:\t\t" + row[2]
        print "* SourcePort:\t\t" + str(row[3])
        print "* DestIP:\t\t\t" + row[4]
        print "* DestPort:\t\t" + str(row[5])
        print "* UrlAccessed:\t\t" + row[6]
        print "* AccessAllowed:\t" + str(row[7])
        print "* CertId:\t\t" + str(row[8])
        print "* PathId:\t\t" + str(row[9])
        pathId = row[9]
        certId = row[8]
    c.close()

    if False == results.infoOnly:
        print ""
        DumpCerts(pathId, certId, results)
    print ""

if __name__ == "__main__":
    main()
```

Appendix H: PITT and PITTv2 Differences

PITTv2 is a complete re-write of PITT but provides much of the same functionality and uses similar user interfaces where possible. PITT implemented several features that were not heavily utilized and were dropped in PITTv2. This section discusses features not present (or substantially altered) in PITTv2 as well as summarizing features unique to PITTv2.

PITT features not in PITTv2

PITTv2 does not provide support for project files and does not interoperate with PITT project files. However, the settings database files used by PITTv2 can support multiple named path settings definitions and multiple security named environment definitions. Settings can be created and named to provide project file-like support within a single settings database. Multiple settings database files can be used as well, as another means of achieving project file-like support.

At present, PITTv2 does not support SCVP.

PITTv2 does not feature a **Single End Entity Path** panel. However, when the **All Paths** panel is used with the **Return first path only** option checked, a similar output is obtained. PITTv2 supports validating multiple certificates read from a folder, where PITT only supported validating a single certificate per operation.

PITTv2 is not open source due to use of streaming ASN.1 parsing code when processing CRLs. A build without streaming support could be prepared if an open source release is desired (albeit with diminished performance). There were few (if any) source contributions resulting from PITT's availability in source form.

PITTv2 does not use native certificate, trust anchor or CRL stores. PITTv2 uses TACT trust anchor stores (which can be managed using TA Store Manager) or file folders for intermediate CA certificates and CRLs.

Features unique to PITTv2

PITTv2 shares much in common with utilities included with TACT, including common file formats and common user interfaces. PITTv2 provides several features that are not present in PITT, including the following:

- Support for multiple platforms (Windows and Linux installers available, Mac binaries available upon request[3])

[3] There are some minor user interface quirks on OSX.

- Render a PKI as a graph
- Determine if a given certificate is present on a given CRL
- Process multiple certificates as part of one validation operation
- Save logs or artifacts from more than one result
- Determine which CAs are connected to current trust anchor set without performing dynamic path discovery (plus additional reports generated from a static PKI graph)
- Avoid cluttering system certificate and CRL stores
- Richer certification path validation logs, including named certificate policies and representation of additional information that influences certification path validation
- PITTv2 does not provide a means of specifying an initial certificate policy set when using CAPI and does not provide a "build-only" option when using CAPI.
- PITTv2 does not provide generation of report summaries.

Appendix K: Default TACT/PITTv2 Configuration Files

The PITTv2 installer packages include the following nine configuration files, which are also included in the TACT installers:

- All_DoD_Approved.tas
- DoD_ECA_Federal.tas
- DoD_ECA.tas
- DoD_only.tas
- FedPolicies.pdb
- All_DoD_Approved.sdb
- DoD_ECA_Federal.sdb
- DoD_ECA.sdb
- DoD.sdb

The All_DoD_Approved.tas file contains all trust anchors from the DOD approved PKIs, as noted here: http://iase.disa.mil/pki-pke/interoperability/index.html. The DoD_ECA_Federal.tas file is a reduction of the complete set that includes the DOD root, the DOD Interoperability Roots, the ECA root and all approved Federal PKI roots. The DoD_ECA.tas file further reduces the set to include only the DOD root and the ECA root. The DoD_only.tas file contains only the DOD root. Each file also includes a DOD trust anchor manager. Each trust anchor other than the DOD root and interoperability roots has an associated excluded name constraint for the DOD DN namespace.

The FedPolicies.pdb file is a certificate policies database containing selected policies from the set of DOD approved PKIs and includes mappings from DOD policies to these policies, where applicable. Federal PKI policies and associated mappings are also included.

Each of the .sdb files references contains four security environment definitions, a single path settings definition and a single TACT resource definition for /. The path settings definition turns on the required explicit policy flag and the enforce trust anchor constraints flags. Each security environment definition references a single trust anchor store (consistent with the name of the security environment) and FedPolicies.pdb. The TACT resource definition varies with the .sdb file. In All_DoD_Approved.sdb, the / resource references the security environment that includes All_DoD_Approved.tas. In DoD_ECA_Federal.sdb, the / resource references the security environment that includes DoD_ECA_Federal.tas. In DoD_ECA.sdb, the / resource references the security environment that includes DoD_ECA.tas. In DoD.sdb, the / resource references the security environment that includes DoD_only.tas

The TACT installers use the All_DoD_Approved.sdb file by default. TACT operators are free to define additional resources that target alternative security environments or to change or remove the restrictions on /.

These files will be updated as new PKIs are approved or PKIs that are currently approved are decommissioned or become no longer approved. Updates will be available on the PKE interoperability site referenced above.

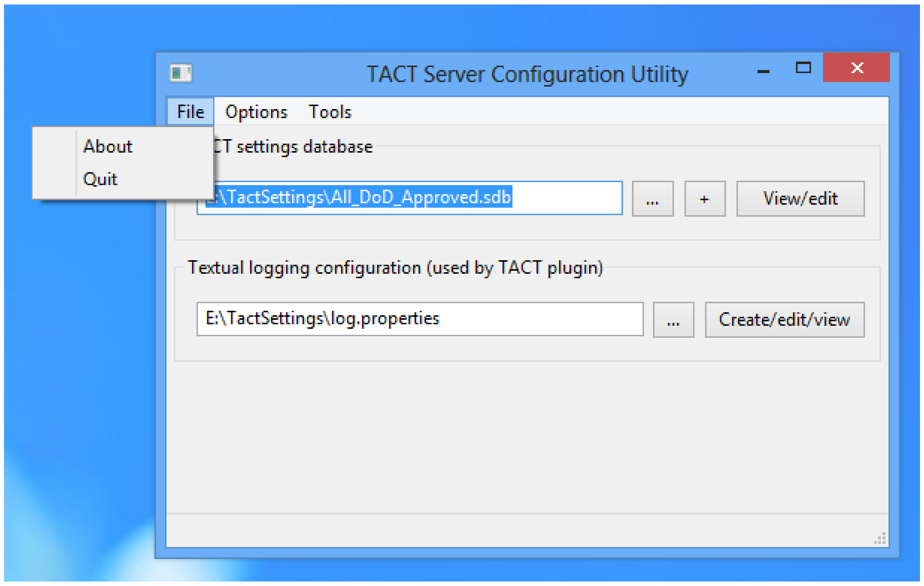
Appendix L: Acronyms

ACL	Access Control List
AIA	Authority Information Access
CA	Certificate Authority
CAPI	Cryptographic Application Programming Interface
CMS	Cryptographic Message Syntax
CSV	Comma-Separated Values
DNS	Domain Name System
DOD	Department of Defense
DOS	Department of State
ECA	External Certification Authority
IIS	Internet Information Services
IRCA	Interoperability Root Certification Authority
PITT	PKI Interoperability Test Tool
PKE	Public Key Enablement
PKI	Public Key Infrastructure
RDN	Relative Distinguished Name
RFC	Request for Comments (Internet Engineering Task Force publication)
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
TA	Trust Anchor
TACT	Trust Anchor Constraint Tool
TactCli	TACT Compliance Assessment Utility
TAMP	Trust Anchor Management Protocol
TASM	Trust Anchor Store Manager
TCA	TACT Compliance Assessment
TLS	Transport Layer Security
TSC	TACT Server Configuration
UPN	User Principal Name
URI	Uniform Resource Identifier
UTC	Coordinated Universal Time
XML	Extensible Markup Language

Appendix N: Windows 8 GUI Appearance

Windows 8 introduced significant changes to the appearance of many graphical controls. It also attempts to enhance tablet/touchscreen behavior and expand tablet support to devices not typically considered "tablets," even if these devices are not equipped with touch sensitive screens.

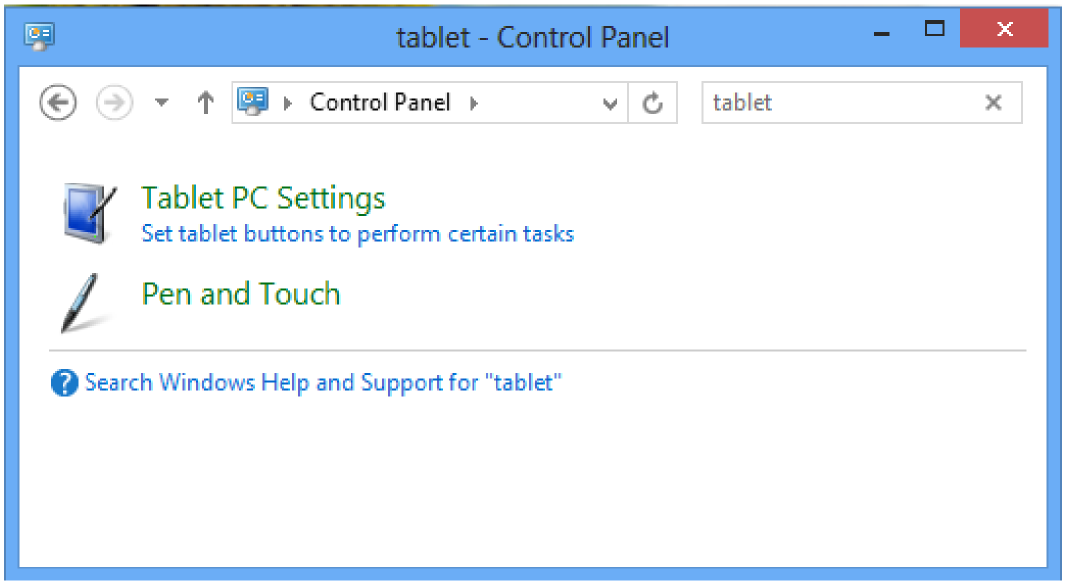
One of these enhancement attempts can cause certain menus to display in an unusual way in many desktop applications, including the TACT GUI administration utilities.


TACT Server Configuration Utility
File
Options
Tools
About
Quit
CT settings database
...
+
View/edit
Textual logging configuration (used by TACT plugin)
...
Create/edit/view

E:\TactSettings\All_DoD_Approved.sdb

E:\TactSettings\log.properties

In order to restore the traditional menu appearance, the "Tablet PC Settings" control panel can be used. Open the control panel and use the search field on the right to search for "tablet."


tablet - Control Panel

Control Panel

tablet

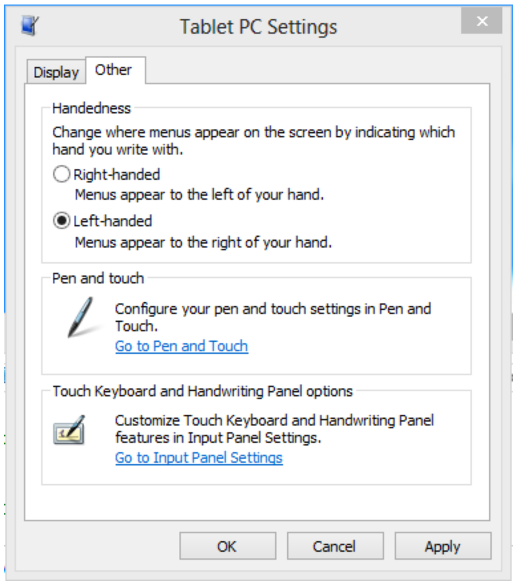
Tablet PC Settings

Set tablet buttons to perform certain tasks

Pen and Touch

Search Windows Help and Support for "tablet"

Within "Tablet PC Settings," go to the "Other" tab and change "Handedness" to "Left-handed."


Tablet PC Settings

Display | Other

Handedness

Change where menus appear on the screen by indicating which hand you write with.

( ) Right-handed
Menus appear to the left of your hand.

(●) Left-handed
Menus appear to the right of your hand.

Pen and touch

Configure your pen and touch settings in Pen and Touch.
Go to Pen and Touch

Touch Keyboard and Handwriting Panel options

Customize Touch Keyboard and Handwriting Panel features in Input Panel Settings.
Go to Input Panel Settings

OK | Cancel | Apply

Click apply. The menus in the TACT GUI tools will then be aligned normally, as will menus in other applications that are impacted by this setting.

Appendix O: Path Builder Differences vs. PITTv1

The certification path discovery implementations in PITTv1 and PITTv2 are quite different. Understanding the structure of the builders may be helpful in some troubleshooting scenarios.

The first difference between the two implementations is the types of sources that are consulted during certification path discovery. PITTv1 can use combinations of the following:

- Microsoft CAPI certificate and CRL stores (current user and/or local machine)
- In-memory cache
- Microsoft CAPI trust anchor and CRL stores (current user and/or local machine)
- A registry-based list of trust anchors, certificates or CRLs
- NSS trust anchor, certificate and CRL stores
- Manually configured LDAP-accessible directory servers
- Manually configured locally-trusted OCSP responders
- Manually configured SCVP responders
- OCSP responders identified in AIA extensions
- CRLs identified in CRL DP extensions (either HTTP or LDAP)
- Certificates identified in AIA or SIA extensions (either HTTP or LDAP)

PITTv2 can use combinations of the following:

- File folder containing certificates
- File folder containing CRLs
- TACT trust anchor store
- Manually configured locally-trusted OCSP responders
- OCSP responders identified in AIA extensions
- CRLs identified in CRL DP extensions (HTTP)
- CRLs identified in CRL DP extensions (LDAP)
- Certificates identified in AIA or SIA extensions (HTTP)
- Certificates identified in AIA or SIA extensions (LDAP)

Both feature ability to blacklist servers, though PITTv1 allows for distinction between HTTP and LDAP. PITTv2 provides means of blacklisting certificates during building, including ignoring all expired certificates. PITTv2 provides means of whitelisting or blacklisting specific certificates for revocation status determination purposes. PITTv2 provides means of configuring granular CRL freshness requirements and grace periods, including the ability to use a grace period only as a last resort. Path validation capabilities are more similar, though PITTv2 supports enforcement of TA-based constraints, algorithm/key size constraints and country code constraints.

Aside from differences in sources and configurability, the certificate path builders included in PITTv1 and PITTv2 are quite different. PITTv1 begins with a set of trust anchor certificates and a target certificate. It then begins path discovery by initially consulting locally available repositories for certificates to form a path from a target certificate to a trust anchor. When more than one certificate is received when querying for an issuer's certificate, a set of heuristics is used to determine which certificate is most likely to result in a valid path. The certificates are sorted based on these heuristics and path discovery continues through each certificate in turn. The builder is essentially a table with the target certificate at the bottom with a changing set of certificates and rows above it. Once a remote resource is required during a path building operations, URIs are freely consulted with an aim of capturing certificates locally to improve build times in the future.

PITTv2 begins with a graph containing trust anchors and locally available certificates. As certificates are added to the graph, edges between nodes are formed when there is an AKID/SKID match or a signature verification relationship. Provided there are no mismatched AKID/SKIDs (i.e., where an AKID references an incorrect SKID), all paths through the graph are cryptographically valid. When a path building operation is begun, a target certificate is attached to the graph and all available paths are discovered. If remote building capabilities are enabled, the URIs captured from each node used to form a path are pushed into a queue for retrieval. Retrieval operations use the uriLastModified.db file written to the certificates or CRLs folder to avoid downloading resources that have not changed since last downloaded.

In PITTv1, network resources are less efficiently used, but paths are discovered in a more definitive order. In PITTv2, network resources are more efficiently used, but the order of discovery can vary. In PITTv1, the time required to build and validate a path is highly influenced by the time required to download certificates and revocation status information. In PITTv2, the time required to build and validate a path is influenced by the time required to download revocation status information.

Appendix P: PITTv1 “Simple store” equivalent

Amongst the differences between PITTv1 and PITTv2 is the lack of “simple” (or registry-based) stores in PITTv2. However, it is possible to achieve a similar functionality using a set of folders and a TA store dedicated to serving as “simple” equivalents. The settings file included with PITTv2 includes support for folders and a TA store for this purpose on Windows: a path settings configuration named “Simple” and a security environment named “Simple Store”. On Linux, a TA store is included but the PITTv2 user must set up the folders that will be used by the “Simple” path settings definition.

As with all TA stores used by PITTv2, the TA Store Manager utility must be used to edit the simple.tas file. When editing a TA store with PITTv2 open, toggle the selected security environment or path settings to force a reload. Make sure to apply and save any changes in TA Store Manager prior to using the updated TA store for testing.

The certificate and CRL stores can be edited using a file browser or through the path settings editor included in PITTv2. Deletions to the certificates file folder will not cause deletions to the currently loaded PITTv2 graph. To reload the graph with fewer CAs, toggle the selected security environment or path settings.