UrbanCode Deploy plugin for MQ Key and Certificate Management

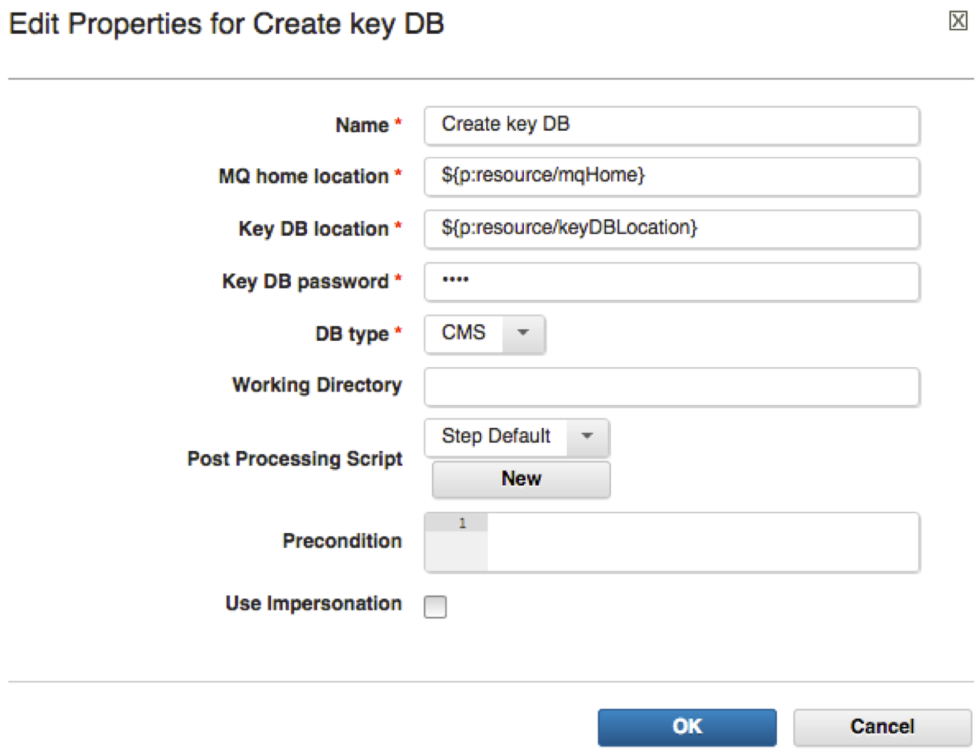
Mark Roberts
UrbanCode Technical Specialist

Functions available in the plugin

- Create a key database
- Backup a key database
- List certificates in the key database
- Add certificates to the key database
- Create a certificate signing request (CSR) from the key database
- Receive a certificate into the key database

Create a key database

- Create a new key database
- MQ Home location is used for access to MQ executables
  - – Defined on a parent resource that specifices information about MQ
- The Key DB location is tored in the field indicated by Key DB location
  - – Default is a resource property ‘keyDBLocation’
- Key DB password is asecure property but a resource property should be used to define it
  - – Past in the value ${p:resource/keyDBPassword}
- DB type is one of CMS, KDB, PKCS 12 or P12

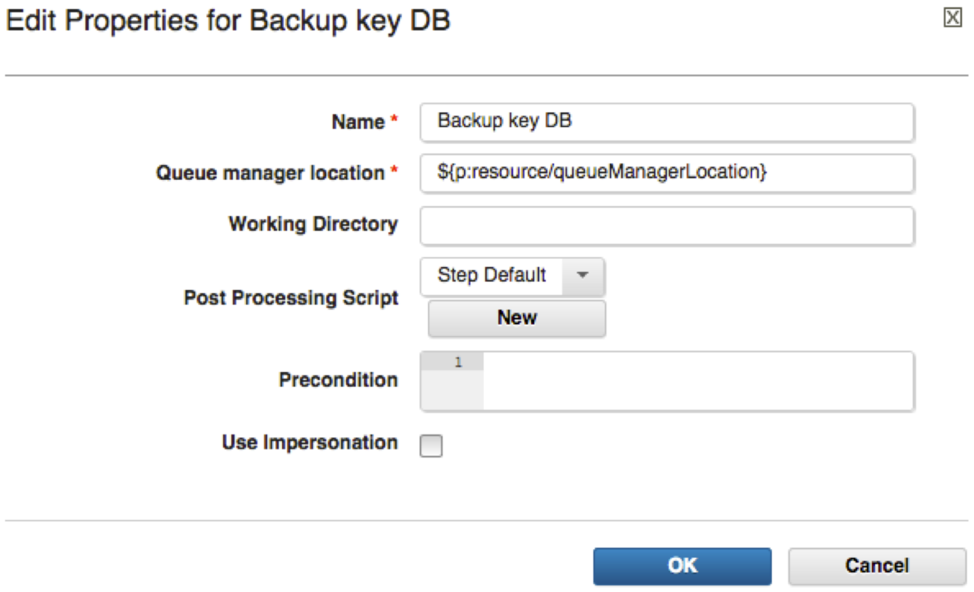


Edit Properties for Create key DB

Name *	Create key DB
MQ home location *	${p:resource/mqHome}
Key DB location *	${p:resource/keyDBLocation}
Key DB password *	••••
DB type *	CMS
Working Directory	
Post Processing Script	Step Default
Precondition	1
Use Impersonation	☐

Backup a key database

- Create a backup of the current key database files
- Current /ssl directory is backed up to /ssl.<date-string>
  – Date string format is dd-MMM-yyyy-HH-mm
- Queue manager location argument is the directory that holds the ssl directory to be backed up

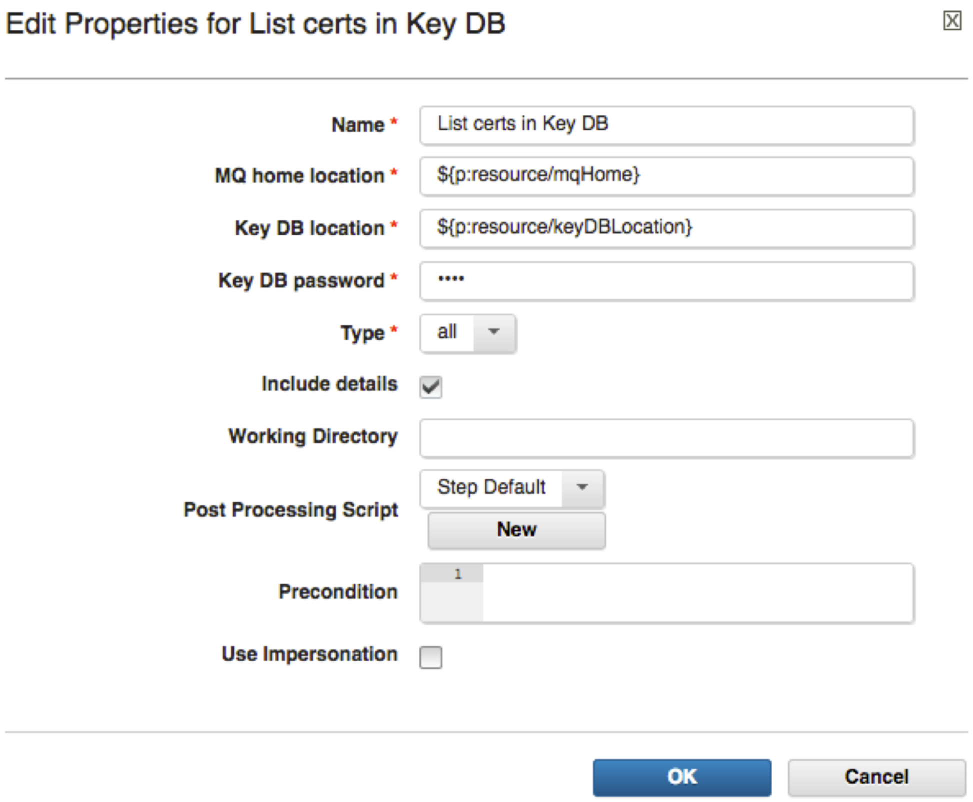


Edit Properties for Backup key DB

Name *	Backup key DB
Queue manager location *	${p:resource/queueManagerLocation}
Working Directory	
Post Processing Script	Step Default ▾ New
Precondition	
Use Impersonation	☐

List certificates in the key database

- Display the certificates in the key database
- MQ Home location is used for access to MQ executables
  - – Defined on a parent resource that specifices information about MQ
- The Key DB location is tored in the field indicated by Key DB location
  - – Default is a resource property ‘keyDBLocation’
- Key DB password is asecure property but a resource property should be used to define it
  - – Past in the value ${p:resource/keyDBPassword}
- Type – select which certificates to display
  - – All, personal or ca

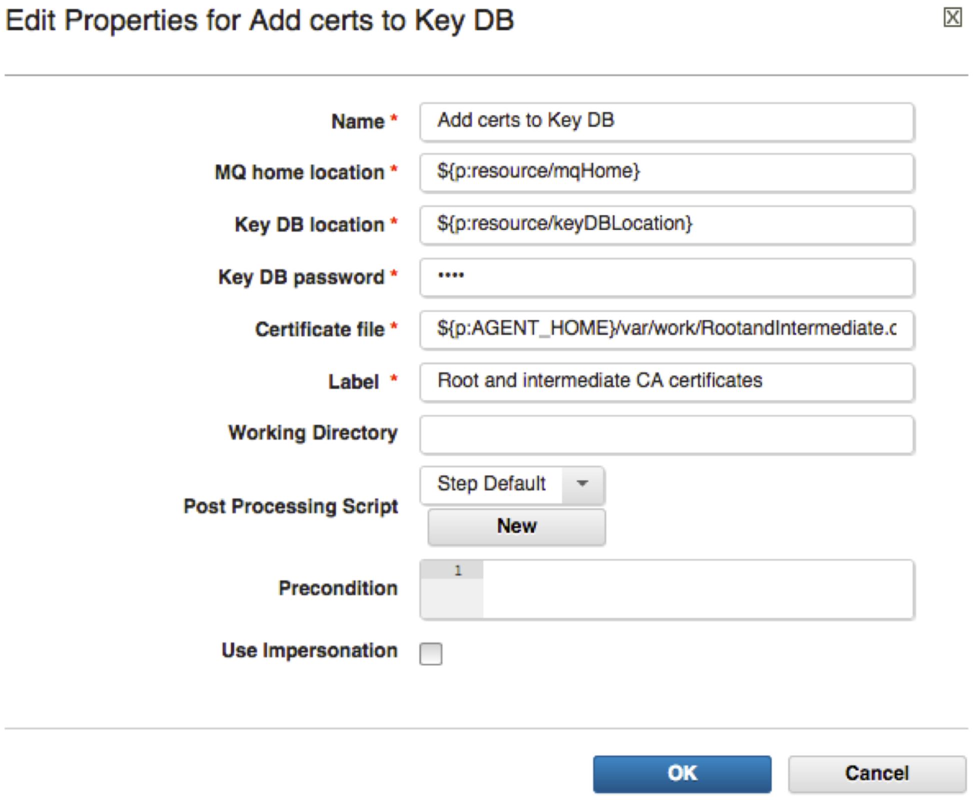


Edit Properties for List certs in Key DB

Name *	List certs in Key DB
MQ home location *	${p:resource/mqHome}
Key DB location *	${p:resource/keyDBLocation}
Key DB password *	••••
Type *	all
Include details	☑
Working Directory	
Post Processing Script	Step Default New
Precondition	1
Use Impersonation	☐

Add certificates to the key database

- Add certificates to the key database
  - May come from a stored file containing root and intermediate certificates
  - May come from the result of a CSR request
- MQ Home location is used for access to MQ executables
  - Defined on a parent resource that specifices information about MQ
- The Key DB location is tored in the field indicated by Key DB location
  - Default is a resource property ‘keyDBLocation’
- Key DB password is asecure property but a resource property should be used to define it
  - Past in the value ${p:resource/keyDBPassword}
- Certificate file
- Label – to help the management of the certificates in the key store

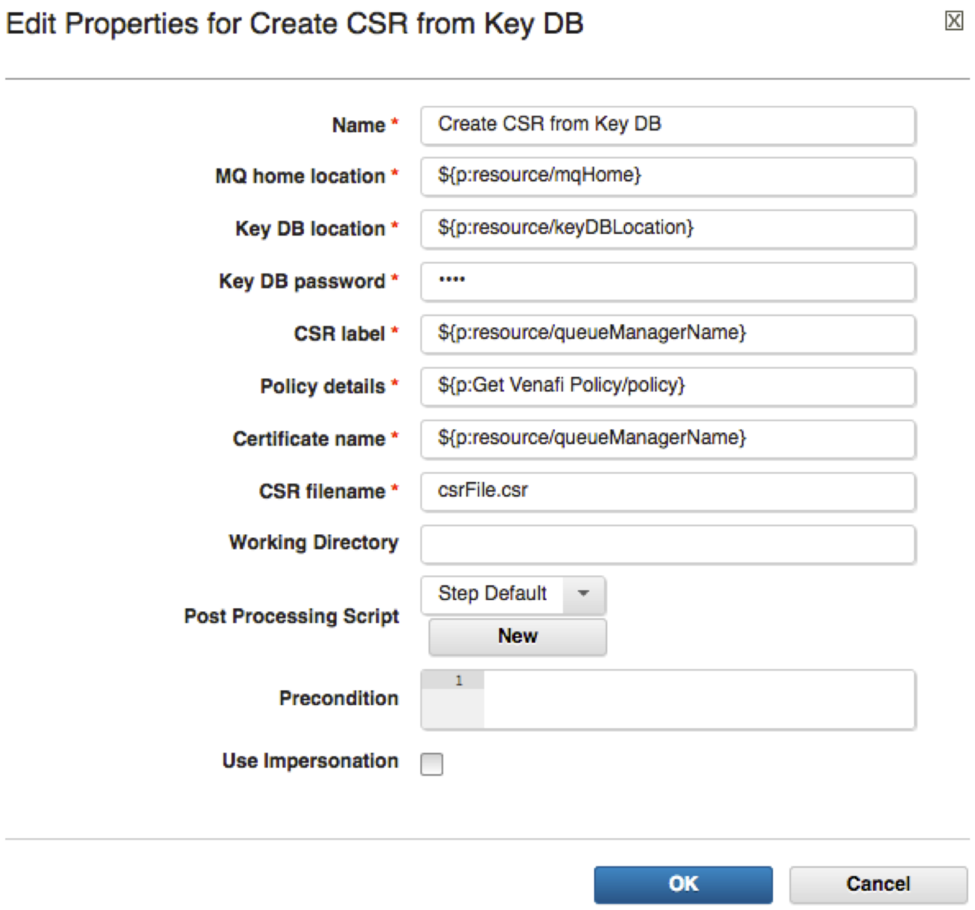


Edit Properties for Add certs to Key DB

Name *	Add certs to Key DB
MQ home location *	${p:resource/mqHome}
Key DB location *	${p:resource/keyDBLocation}
Key DB password *	••••
Certificate file *	${p:AGENT_HOME}/var/work/RootandIntermediate.c
Label *	Root and intermediate CA certificates
Working Directory	
Post Processing Script	Step Default
Precondition	
Use Impersonation	☐

Create a certificate signing request (CSR) from the key database

- Create a CSR to be submitted to a certificate provider
- MQ Home location is used for access to MQ executables
  - – Defined on a parent resource that specifices information about MQ
- The Key DB location is tored in the field indicated by Key DB location
  - – Default is a resource property ‘keyDBLocation’
- Key DB password is asecure property but a resource property should be used to define it
  - – Past in the value ${p:resource/keyDBPassword}
- CSR label – a label to describe the certificate
- Policy details – JSON formatted policy data (see next slide)
- Certificate name
- CSR filename



Edit Properties for Create CSR from Key DB ☒

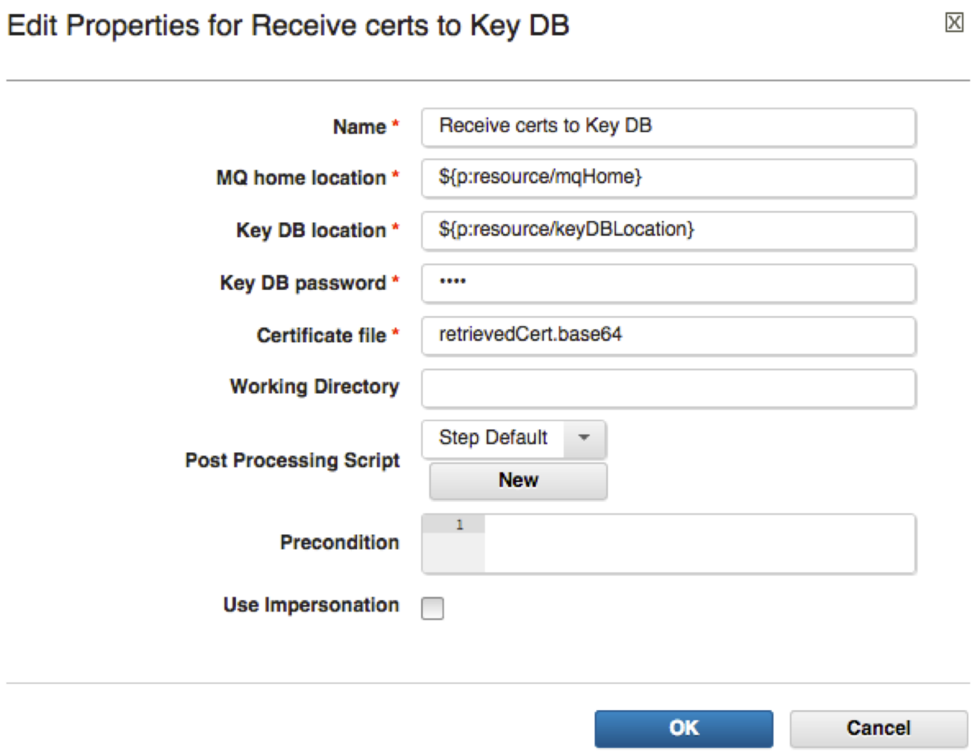
Name *	Create CSR from Key DB
MQ home location *	${p:resource/mqHome}
Key DB location *	${p:resource/keyDBLocation}
Key DB password *	••••
CSR label *	${p:resource/queueManagerName}
Policy details *	${p:Get Venafi Policy/policy}
Certificate name *	${p:resource/queueManagerName}
CSR filename *	csrFile.csr
Working Directory	
Post Processing Script	Step Default ▾ New
Precondition	1
Use Impersonation	□

Create a certificate signing request (CSR) from the key database

- Policy details :
  - “name”:”value” pairs comma separated – for example :
  - {"city":”Birmingham","state":"West Midlands","country":"UK","organization":"IBM UK Ltd","organizationalUnit":"Hybrid Cloud Division","keyAlgorithm":"RSA","keyBitStrength":"2048","managementType":"Monitoring","manualCSR":"ServiceGenerated","certificateAuthority":"\\VED\\Policy\\Certificate authorities\\MS CA","policyDN":"\\VED\\Policy\\DevOps_Workshop"}

Receive a certificate into the key database

- Add received certificate data into the keystore
- Note that they request for the certificates must have come from a CSR generated from the key store using the step ‘Create a certificate signing request (CSR) from the key database’
- MQ Home location is used for access to MQ executables
  - – Defined on a parent resource that specifices information about MQ
- The Key DB location is tored in the field indicated by Key DB location
  - – Default is a resource property ‘keyDBLocation’
- Key DB password is asecure property but a resource property should be used to define it
  - – Past in the value ${p:resource/keyDBPassword}
- Certficate file – containing the generated certificate



Edit Properties for Receive certs to Key DB

Name *	Receive certs to Key DB
MQ home location *	${p:resource/mqHome}
Key DB location *	${p:resource/keyDBLocation}
Key DB password *	••••
Certificate file *	retrievedCert.base64
Working Directory	
Post Processing Script	Step Default New
Precondition	1
Use Impersonation	□