

co.za Policy Engine

Michael O'Connell

Software Architect

Domain Name Services (Pty) Ltd

Business Domain

- Policy is a language of Business
- Policy is ever changing
- Not interested in technical implementation
- Technical implementation should not hamper policy development

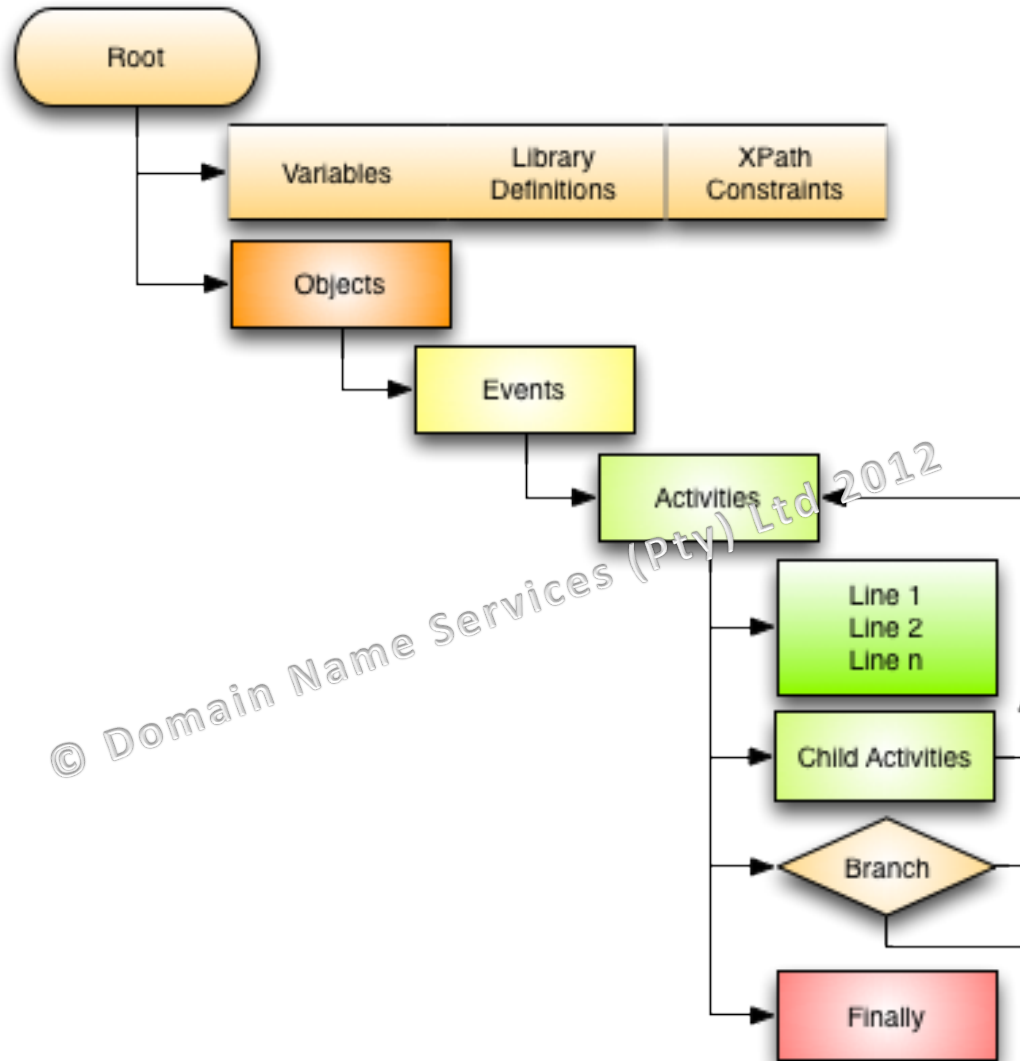
Technical Domain

- Efficiently/Effectively/Flexibly accommodate Business needs
 - Hierarchical Procedural Policy Structure
 - Dynamic library based functionality
 - Unix-paradigm of micro-commands
- Built-in Engine for on-the-fly reloading
 - Policy based in XML structure
 - DB/File system based

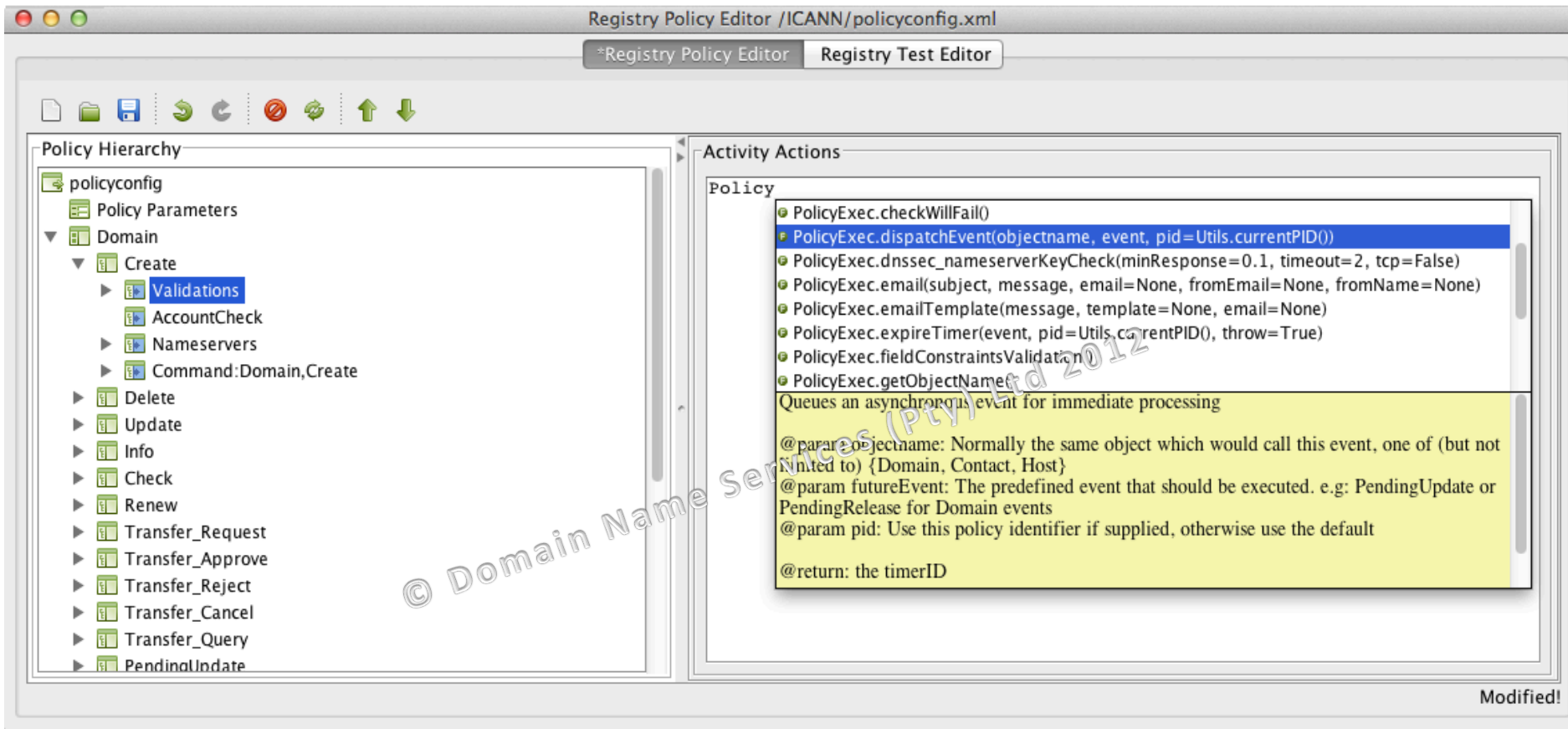
Capabilities

- Map business rules directly to policy
- Policy variable declaration
- 3rd Party Library declaration
- XML XPath based validation definitions
- UI based Policy merging
- Contextual popup helpers
- Language agnostic

Structure



User Interface



Registry Policy Editor /ICANN/policyconfig.xml

*Registry Policy Editor Registry Test Editor

Policy Hierarchy

- policyconfig
 - Policy Parameters
 - Domain
 - Create
 - Validations
 - AccountCheck
 - Nameservers
 - Command:Domain,Create
 - Delete
 - Update
 - Info
 - Check
 - Renew
 - Transfer_Request
 - Transfer_Approve
 - Transfer_Reject
 - Transfer_Cancel
 - Transfer_Query
 - PendinalUpdate

Activity Actions

Policy

- PolicyExec.checkWillFail()
- PolicyExec.dispatchEvent(objectname, event, pid=Utils.currentPID())
- PolicyExec.dnssec_nameserverKeyCheck(minResponse=0.1, timeout=2, tcp=False)
- PolicyExec.email(subject, message, email=None, fromEmail=None, fromName=None)
- PolicyExec.emailTemplate(message, template=None, email=None)
- PolicyExec.expireTimer(event, pid=Utils.currentPID(), throw=True)
- PolicyExec.fieldConstraintsValidation()
- PolicyExec.getObjectNames()

Queues an asynchronous event for immediate processing

@param objectname: Normally the same object which would call this event, one of (but not limited to) {Domain, Contact, Host}

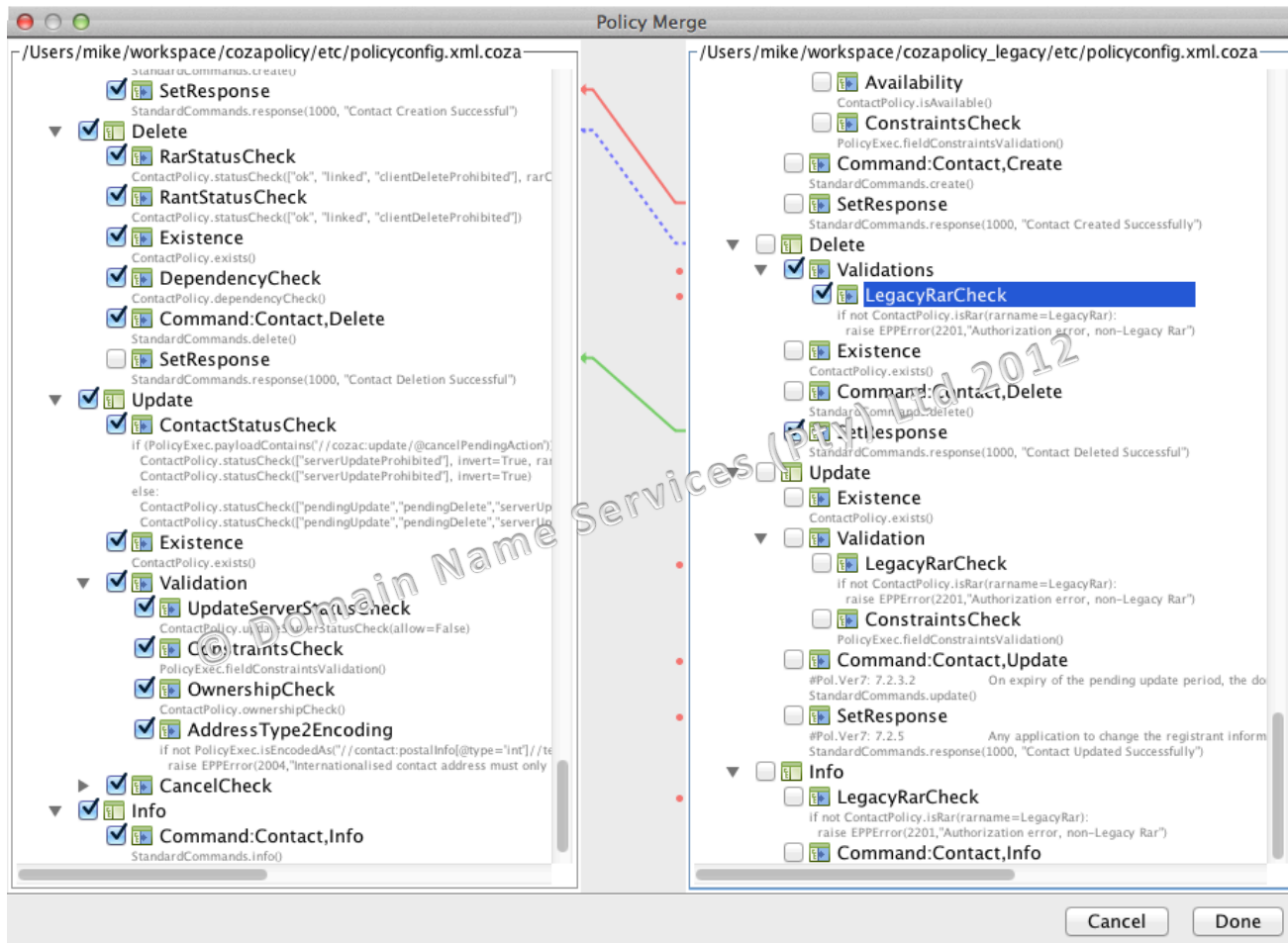
@param futureEvent: The predefined event that should be executed. e.g: PendingUpdate or PendingRelease for Domain events

@param pid: Use this policy identifier if supplied, otherwise use the default

@return: the timerID

Modified!

User Interface contd. - Merging



The screenshot displays the 'Policy Merge' dialog box, comparing two policy configurations. The left pane shows the current policy configuration, and the right pane shows the legacy policy configuration. The 'LegacyRarCheck' is highlighted in the right pane.

Left Pane (Current Policy):

- SetResponse**
StandardCommands.response(1000, "Contact Creation Successful")
- Delete**
 - RarStatusCheck**
ContactPolicy.statusCheck(["ok", "linked", "clientDeleteProhibited"], rarC
 - RantStatusCheck**
ContactPolicy.statusCheck(["ok", "linked", "clientDeleteProhibited"])
 - Existence**
ContactPolicy.exists()
 - DependencyCheck**
ContactPolicy.dependencyCheck()
 - Command:Contact,Delete**
StandardCommands.delete()
 - SetResponse**
StandardCommands.response(1000, "Contact Deletion Successful")
- Update**
 - ContactStatusCheck**
if (PolicyExec.payloadContains("/coza: update/@cancelPendingAction"),
ContactPolicy.statusCheck(["serverUpdateProhibited"], invert=True, ra
ContactPolicy.statusCheck(["serverUpdateProhibited"], invert=True)
else:
ContactPolicy.statusCheck(["pendingUpdate", "pendingDelete", "serverUp
ContactPolicy.statusCheck(["pendingUpdate", "pendingDelete", "serverUp
 - Existence**
ContactPolicy.exists()
 - Validation**
 - UpdateServerStatusCheck**
ContactPolicy.updateServerStatusCheck(allow=False)
 - ConstraintsCheck**
PolicyExec.fieldConstraintsValidation()
 - OwnershipCheck**
ContactPolicy.ownershipCheck()
 - AddressType2Encoding**
if not PolicyExec.isEncodedAs["/contact.postalInfo[@type='int']//te
raise EPPError(2004, "Internationalised contact address must only
 - CancelCheck**
- Info**
 - Command:Contact,Info**
StandardCommands.info()

Right Pane (Legacy Policy):

- Availability**
ContactPolicy.isAvailable()
- ConstraintsCheck**
PolicyExec.fieldConstraintsValidation()
- Command:Contact,Create**
StandardCommands.create()
- SetResponse**
StandardCommands.response(1000, "Contact Created Successfully")
- Delete**
 - Validations**
 - LegacyRarCheck**
if not ContactPolicy.isRar(rarname=LegacyRar):
raise EPPError(2201, "Authorization error, non-Legacy Rar")
 - Existence**
ContactPolicy.exists()
 - Command:Contact,Delete**
StandardCommands.delete()
 - SetResponse**
StandardCommands.response(1000, "Contact Deleted Successful")
 - Update**
 - Existence**
ContactPolicy.exists()
 - Validation**
 - LegacyRarCheck**
if not ContactPolicy.isRar(rarname=LegacyRar):
raise EPPError(2201, "Authorization error, non-Legacy Rar")
 - ConstraintsCheck**
PolicyExec.fieldConstraintsValidation()
 - Command:Contact,Update**
#Pol.Ver7: 7.2.3.2 On expiry of the pending update period, the do
StandardCommands.update()
 - SetResponse**
#Pol.Ver7: 7.2.5 Any application to change the registrant inform
StandardCommands.response(1000, "Contact Updated Successfully")
 - Info**
 - LegacyRarCheck**
if not ContactPolicy.isRar(rarname=LegacyRar):
raise EPPError(2201, "Authorization error, non-Legacy Rar")
 - Command:Contact,Info**

gTLD Launch Phase

Registry Policy Editor /ICANN/policyconfig.xml

Registry Policy Editor Registry Test Editor

Policy Hierarchy

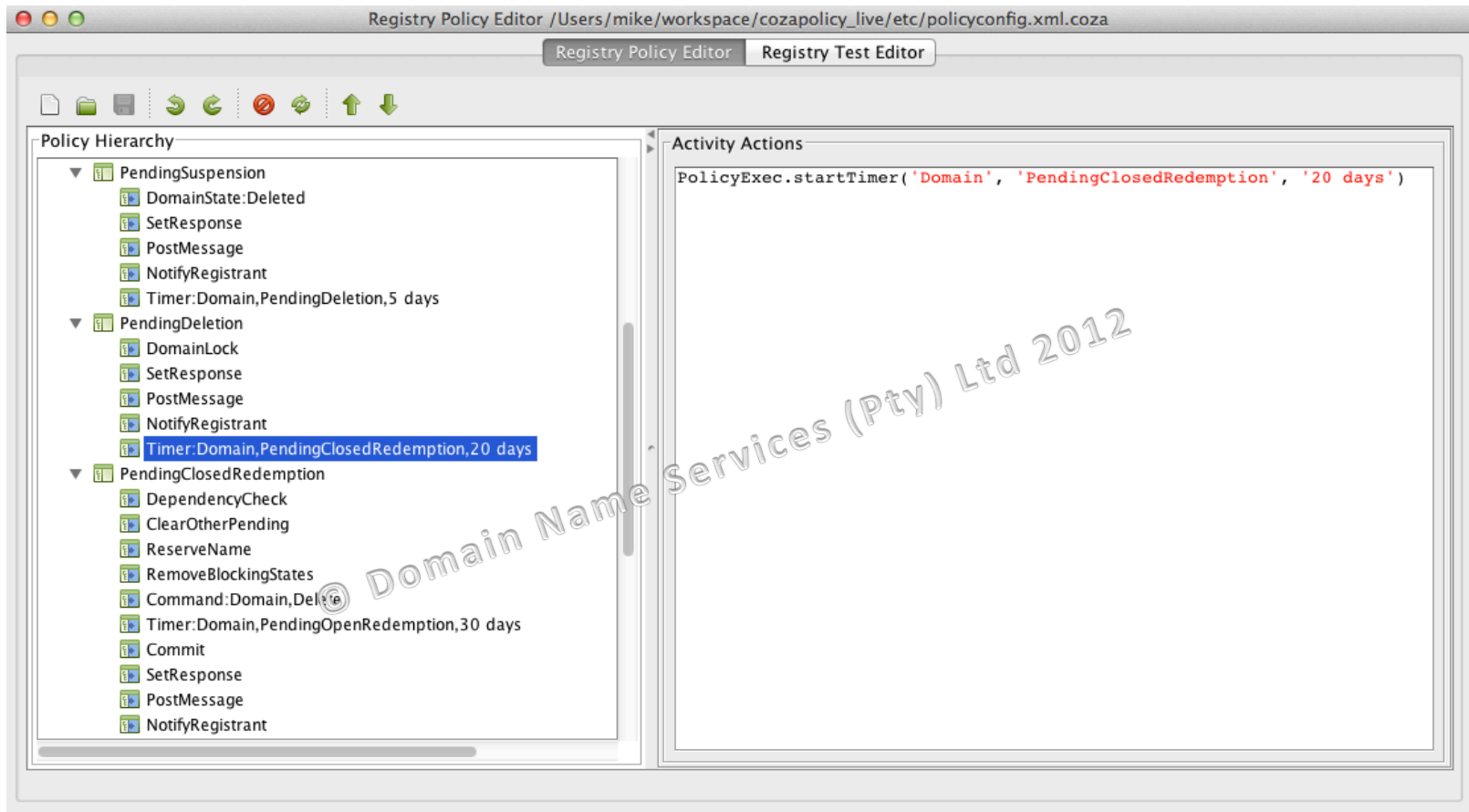
- policyconfig
 - Policy Parameters
 - Domain
 - Create
 - CheckSunrisePeriod
 - PolicyExec.isBeforeDate('2013-05-01')
 - Sunrise
 - Other
 - PolicyExec.isWithinDate('2013-05-01', '2013-06-01')
 - Landrush
 - ReserveName
 - NotifyApplicant
 - Normal
 - Validations
 - Command:Domain,Create
 - Delete
 - Update
 - Info

Activity Actions

```
# Dates are for example only
if PolicyExec.isWithinDate('2013-03-01', '2013-04-01'):
    SunrisePolicy.reserveName('African')
elif PolicyExec.isWithinDate('2013-04-01', '2013-05-01'):
    SunrisePolicy.reserveName('International')

SunrisePolicy.issueTrademarkClearance()
SunrisePolicy.issueToken()
```


co.za Closed Redemption



The screenshot shows the Registry Policy Editor interface. The title bar indicates the file path: Registry Policy Editor /Users/mike/workspace/cozapolicy_live/etc/policyconfig.xml.coza. The interface has two tabs: "Registry Policy Editor" (selected) and "Registry Test Editor".

The left pane, titled "Policy Hierarchy", shows a tree structure of policy actions:

- ▼ PendingSuspension
 - DomainState:Deleted
 - SetResponse
 - PostMessage
 - NotifyRegistrant
 - Timer:Domain,PendingDeletion,5 days
- ▼ PendingDeletion
 - DomainLock
 - SetResponse
 - PostMessage
 - NotifyRegistrant
 - Timer:Domain,PendingClosedRedemption,20 days
- ▼ PendingClosedRedemption
 - DependencyCheck
 - ClearOtherPending
 - ReserveName
 - RemoveBlockingStates
 - Command:Domain,Delete
 - Timer:Domain,PendingOpenRedemption,30 days
 - Commit
 - SetResponse
 - PostMessage
 - NotifyRegistrant

The right pane, titled "Activity Actions", contains the following code snippet:

```
PolicyExec.startTimer('Domain', 'PendingClosedRedemption', '20 days')
```

A large watermark "© Domain Name Services (Pty) Ltd 2012" is overlaid diagonally across the center of the screenshot.

Registry Policy Editor /Users/mike/workspace/cozapolicy_live/etc/policyconfig.xml.coza

Registry Policy Editor Registry Test Editor

Policy Hierarchy

- PostMessage
- NotifyRegistrant
- Timer:Domain,PendingClosedRedemption,20 days
- ▼ PendingClosedRedemption
 - DependencyCheck
 - ClearOtherPending
 - ReserveName
 - RemoveBlockingStates
 - Command:Domain,Delete
 - Timer:Domain,PendingOpenRedemption,30 days
 - Commit
 - SetResponse
 - PostMessage
 - NotifyRegistrant
- ▼ PendingOpenRedemption
 - ▼ CheckAuctionToken
 - ▼ AuctionPolicy.checkToken() # Determine if there is an open
 - ▼ WinnerDetermined
 - GenerateCreate
 - Command:Domain,Create
 - ReleaseReservation
 - ▼ Indeterminate
 - ReleaseReservation

Activity Actions

```
PolicyExec.startTimer('Domain', 'PendingOpenRedemption', '30 days')
```

© Domain Name Services (Pty) Ltd 2012

Q & A

Michael O'Connell

Software Architect

Domain Name Services (Pty) Ltd

mike@dnservices.co.za

www.registry.net.za

www.dotafrica.org