

2011_06_07_AVKS_Tivoli Provisioning Manager (TPM) - APIs and WebServices

 Tivoli® software



Agenda

- Webservice Overview
- TPM Server Configuration for Webservices
- Webservice Client Installation
- Usage – Command Line
- Development Environment Setup
- Programmatical API
- Programmatical API WSRF
- CLI with password encryption
- Code Deep Dive
- Troubleshooting I
- Troubleshooting II
- Reference Materials

Webservice Overview

- WSRF (Web Services Resource Framework) defines a family of specifications for accessing stateful resources using webservices. It is a set of specifications that includes WS-ResourceProperties, WS-ResourceLifeTime, WS-BaseFaults, etc

- TPM webservices built on a wsrf-lite framework to manage stateless and stateful webservices
 - ▶ Stateless – a family of webservices for one time invocation operations. E.g. Execute a deployment request, retrieve status of a de request, get de logs, createByXmlImport, etc
 - ▶ Stateful – for managing TPM datacentermodel. User can create, delete, get properties, update properties, etc. Eg. Servers, SoftwareModules, SRTs, etc

- TPM 7.x does not use REST APIs, TPM 8.1 will use REST APIs
 - ▶ Any REST APIs in Tpaee are not exploited in TPM 7.x. The officially supported remote API interface is the WSRF webservices

TPM Server Configuration for Webservices

- Webservices Port
 - ▶ \$TIO_HOME/lwi/conf/overrides/TPMconfig.properties
 - com.ibm.osg.webcontainer.port=8777
 - com.ibm.pvc.webcontainer.port.secure=8778

- Ws-security
 - ▶ \$TIO_HOME/config/user-factory.xml
 - <ws-security>true</ws-security>
 - <ws-security-role>TPWEBSERVICEUSER</ws-security-role>
 - <authentication-realm>...</authentication-realm>

- SOAP Messages (WSRF framework) Logging
 - ▶ Add to \$TIO_HOME/lwi/conf/overrides/TPMconfig.properties
 - logredirector.level=DEBUG | INFO | WARNING | ERROR
 - Outputs to \$TIO_HOME/lwi/logs
 - trace-log.xml & error-log.xml

Webservice Client Installation

- TPM Webservice clients are installed to
 - ▶ \$TIO_HOME/soapclient
 - tpmlteSoap
 - tcwsrfclient
 - ▶ Copy dir to any remote machine to run client from that machine
 - Update script files and replace references to \$JAVA_HOME or set this environment variable on the new client
 - Java is the only prerequisite on client
 - No further setup is required

- For java applications invoking TPM webservices, client libs can be found in the subdirectories
 - tpmlteSoap/lib and tcwsrfclient/lib
 - Proxies and stubs are provided. Use these as they have implementations of the ws* specs required to invoke TPM webservices

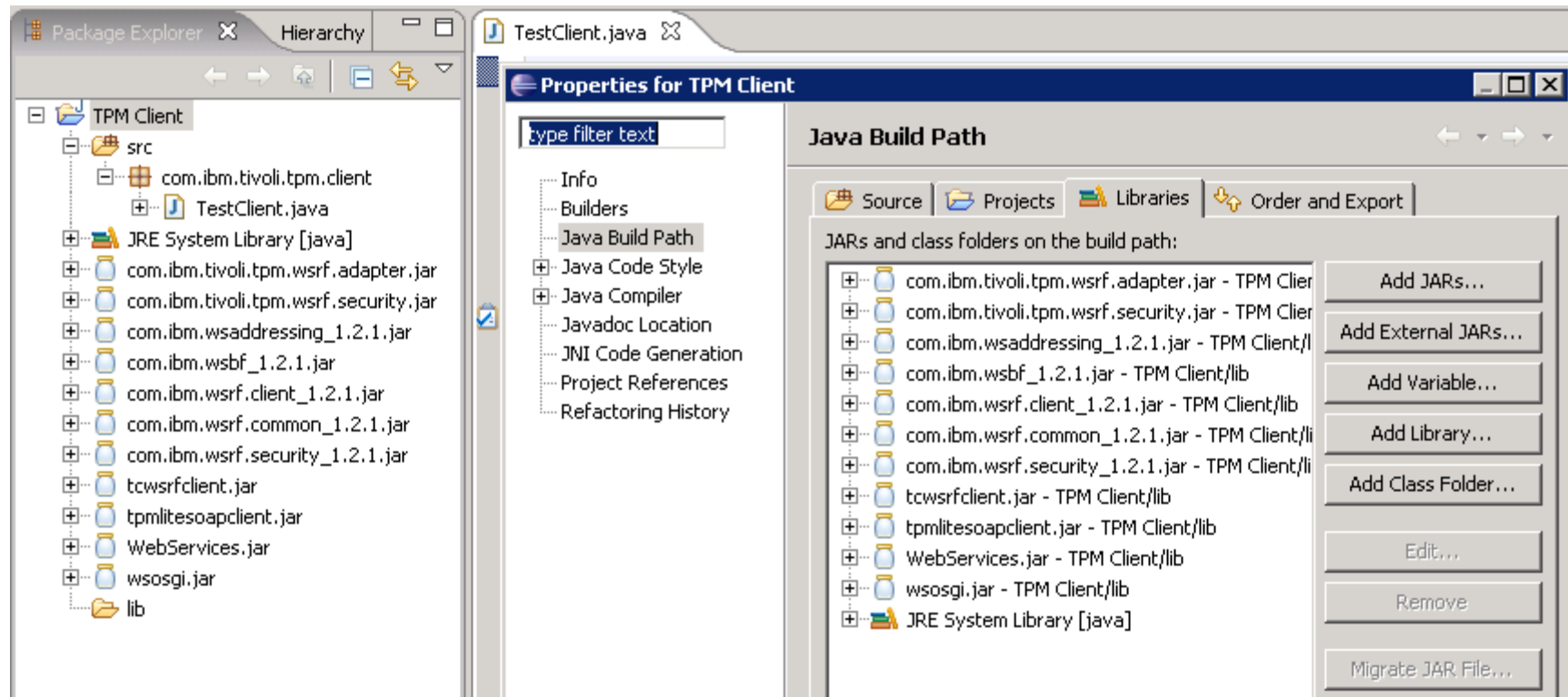
Usage – Command Line

- Stateless Webservices
 - ▶ \$TIO_HOME/soapclient/tpmlteSoap
 - Main script is soapcli.sh
 - `./soapcli.sh <username> <password> <wsdl> <operation> <parameter_list>`
 - Eg. `./soapcli.sh maxadmin password http://localhost:8777/ws/pid/TpmLiteSoapService?wsdl executeDeploymentRequest String_Concatenate String_1=aaa,String_2=bbb`
 - List of wsdl's and operations can be found in the infocenter

- WSRF Webservices
 - ▶ \$TIO_HOME/soapclient/tcwsrfclient
 - Sample menu driven command client
 - `./tcwsrfclient.sh <hostname:port> <username> <password>`
 - Eg. `./tcwsrfclient.sh localhost:8777 maxadmin password`
 - Navigate through the sample to manage TPM

Development Environment Setup

- Install an IDE, eg. RAD, RSA, Eclipse
 - Create a Project
 - Include all libs from \$TIO_HOME/soapclient/tcwsrfclient/lib in classpath
 - Write your client application



Programmatical API

- **Must use proxy and stubs provided by TPM**

- Stateless Webservices
 - ▶ Proxies available
 - TpmLiteSoapServiceProxy
 - CredentialsManagerServiceProxy
 - SPOfferingServiceProxy
 - SPSubscriptionServiceProxy
 - EffectiveModeServiceProxy
 - OperationsModeServiceProxy
 - FaultManagementServiceProxy
 - RecommendationsServiceProxy
 - ResourceInformationServiceProxy
 - TpmEventBrokerServiceProxy

 - ▶ Reference -> SOAP services

 - ▶ Reference -> SOAP services -> SOAP scripts
 - All the comands listed here are available method calls from proxy

Programmatical API cont...

- ▶ Usage
 - Instantiate proxy
 - Invoke method

- ▶ Infocenter Sample
 - Reference -> SOAP services -> Running SOAP from applications

- ▶ Javadocs
 - Reference -> SOAP services->SOAP services API reference

```
public static void main(String args[]) throws Exception {  
  
    String ip = "localhost:8777";  
    TpmLiteSoapServiceProxy proxy = new TpmLiteSoapServiceProxy(ip, "maxadmin", "maxadmin");  
  
    // Invoke workflow  
    String workflowName = "String_Concatenate";  
    String requestParameters = "String_1=aaa,String_2=bbb";  
    long requestId= proxy.executeDeploymentRequest(workflowName, requestParameters);  
    System.out.println("request id = " + requestId);  
  
    // Check status  
    int status= proxy.findDeploymentStatus(requestId);  
    System.out.println("status = " + status);  
}
```

Programmatical API WSRF

- Stateful Webservices
 - ▶ Include all libs from \$TIO_HOME/soapclient/tcwsrfclient/lib in classpath
 - TCWSRFClientProxy

 - ▶ Reference -> Web services -> WSRF services overview

 - ▶ Usage
 - Instantiate proxy
 - Call CreateWSResource method
 - Call findReferenceByProperty to assign identity (for calls requiring identity, eg. getProperty)
 - Invoke methods
 - Call destroyWSResource method

 - ▶ Infocenter Sample
 - Reference -> Web services -> Running WSRF from applications -> WSRF proxy code
 - Functional sample can be run as-is

 - ▶ Javadocs
 - Reference -> Web services-> Web services API reference

CLI with password encryption

- TPM webservice command line syntax exposes password in plain text (e.g. soapcli.sh)
- Recommendation:
 - TPM has published the APIs for customers to write their own clients
 - soapcli.sh is itself an out-of-the-box client which uses the TPM published APIs
 - Write a client to:
 - Read password from a file
 - Decrypt password using decryption technique of customers design
 - Invoke TPM webserivces

```
public static void main(String[] args) {
    try {
        // Read soapcli properties from a file
        Properties properties = new Properties();
        properties.load(new FileInputStream("c:/soapcli.properties"));

        // Get non encrypted properties
        String tpmHostname = properties.getProperty("hostname");
        String tpmUsername = properties.getProperty("username");

        // Get password and decrypt using algorithm of your design
        String encryptedPassword = properties.getProperty("password");
        String decryptedPassword = decrypt(encryptedPassword);

        TpmLiteSoapServiceProxy proxy = new TpmLiteSoapServiceProxy(tpmHostname, tpmUsername, decryptedPassword);
        proxy.executeDeploymentRequest("String_Concatenate", "String_1=aaa,String_2=bbb");
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Code Deep Dive

- Most used Classes
 - ▶ Stateless
 - TpmLiteSoapServiceProxy
 - <http://publib.boulder.ibm.com/infocenter/tivihelp/v38r1/topic/com.ibm.tivoli.tpm.soapjdoc.doc/com/ibm/tivoli/soap/proxy/TpmLiteSoapServiceProxy.html>
 - ▶ WSRF
 - TCWSRFClientProxy
 - <http://publib.boulder.ibm.com/infocenter/tivihelp/v38r1/topic/com.ibm.tivoli.tpm.wsjdoc.doc/com/ibm/tivoli/tpm/wsrf/tcwsrfresource/webservice/proxy/TCWSRFClientProxy.html>
 - ResourcePropertyUtil – Xml helper
 - TCWSRFResource – Generic object holder
- Use Cases
 - ▶ Run a workflow
 - Code sample
 - ▶ Get status
 - Code sample
 - ▶ Retrieve a Server and Server information
 - Code sample
 - ▶ Software Installation
 - Code Sample
 - ▶ Discovery – template creation and execution
 - Code Sample

Troubleshooting I

- **Connection Failed**
 - ▶ Make sure webservices started
 - Check that webservice is listening <http://localhost:8777/ws/pid/TpmLiteSoapService>
- **Resource no longer valid**
 - ▶ Check that client machine and TPM server date/time are no more than 1hr apart
- **Logs**
 - ▶ \$TIO_LOGS/console.log
 - ▶ \$TIO_HOME/lwi/logs
 - trace-log.xml & error-log.xml
- **nullPointerExceptions**
 - ▶ Verify username/password are correct

Troubleshooting II

- **Advanced Troubleshooting (Manually starting TPM engines)**
 - ▶ Stop TPM engines
 - Go to \$TIO_HOME/tools
 - Run `./tio.sh stop -t`

 - ▶ Go to \$TIO_HOME/lwi/bin
 - Run `./lwistart.sh -clean`

 - ▶ When TPM engines has finished starting, you will get an osgi console and something like:
 - eTPM 108 adapter(s) registered
 - eTPM 108 webservice(s) registered
 - eTPM 12 stateless adapter(s) registered
 - eTPM 12 stateless webservice(s) registered

 - ▶ Now you can attempt to access the webservice. Any errors should output to the osgi console

 - ▶ When you are done, type close at the osgi prompt
 - `osgi> close`

Reference Materials

- **TPM Infocenter**
 - ▶ Reference
 - Webservices & SOAP services sections

- **Specifications**
 - ▶ WS-Resource
 - http://docs.oasis-open.org/wsrf/wsrf-ws_resource-1.2-spec-cs-01.pdf2
 - ▶ WS-ResourceLifeTime
 - http://docs.oasis-open.org/wsrf/wsrf-ws_resource_lifetime-1.2-spec-cs-01.pdf3
 - ▶ WS-ResourceProperties
 - http://docs.oasis-open.org/wsrf/wsrf-ws_resource_properties-1.2-spec-cs-01.pdf4
 - ▶ WS-BaseFault
 - http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-cs-01.pdf5
 - ▶ WS-Addressing
 - <http://www.w3.org/TR/2005/CR-ws-addr-core-200508176>

Backup Slides

Tivoli software



