

# Implementing the mediation between departmental services consumers and 3<sup>rd</sup> party providers

---

This document is intended for users who want to deploy scenario artifacts from the Downloads section. The demo will be published from [http://www.ibm.com/isv/marketing/saas/demo\\_series.htm](http://www.ibm.com/isv/marketing/saas/demo_series.htm) for this scenario.

## 1. Pre-requisites and Services Infrastructure setup

Software/Hardware requirements for building the internal mediation proxy layer:

1. WebSphere DataPower Appliance –XI50
2. Tivoli Usage and Accounting Manager 7.1.
3. WebSphere Service Registry and Repository 6.1.
4. WebSphere Process Server 6.1 (also possible to use just WebSphere Application Server v 6.1)

Software requirements for running example departmental business process which need to use 3<sup>rd</sup> party services:

1. WebSphere Process Server 6.1.

Pre-requisites configuration:

- Web services implementation for Expo credit check service is deployed on the WebSphere Process Server. The ear file for this service, ExpoCreditCheckService.ear, is attached to the demo page.
- Credit Score database is created using ExpoCreditScore\_SampleSchema.ddl provided in Downloads section.
- Data source with JNDI name jdbc/BankDST1 is created on the same server where Expo Credit Score Service is deployed. For more information on how to create Data Source, please have a look at following infocenter document link: [http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.base.doc/info/aes/ae/tdat\\_ccrtpds.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.base.doc/info/aes/ae/tdat_ccrtpds.html)
- Folder structure C:/Test is created with Java Key Store expostore.jks inside it, provided in the Downloads section. If you want to use some other path for storing the key store, same needed to be changed in the WS-Binding configuration of Expo Credit Check Service. Please refer to the following developer works article for same: [http://www.ibm.com/developerworks/websphere/techjournal/0606\\_singh/0606\\_singh.html](http://www.ibm.com/developerworks/websphere/techjournal/0606_singh/0606_singh.html)

## 2. Configure the JK Financial's internal service registry using WebSphere Service Registry and Repository (WSRR)

1. Configure a custom user interface in WSRR.
  - a. Open ExpoCreditScore.mod\_ICreditServicesExport1.wsdl in a text editor and change the SOAP address hostname and port number (as shown in Figure 1 below), to the server hostname and port number where this Expo Credit Score Service is deployed.

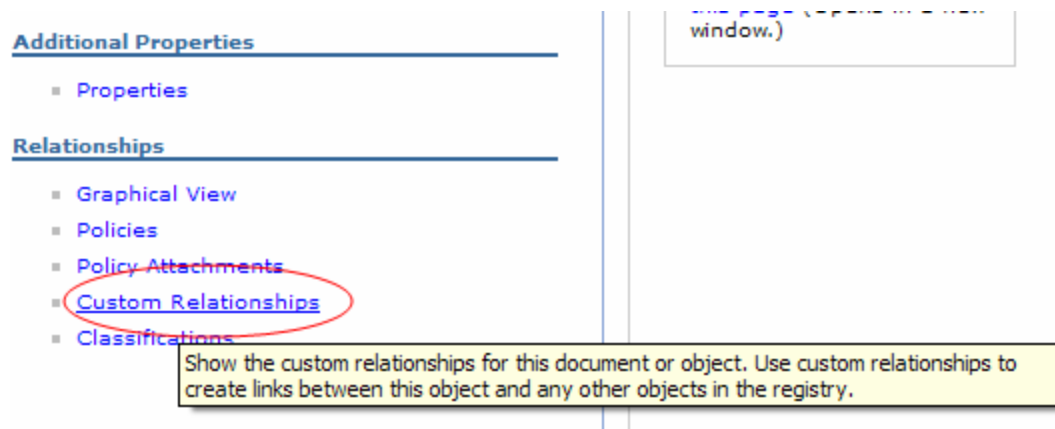
```

</wsdl:binding>
<wsdl:service name="ICreditServicesExport1_ICreditServicesHttpService">
  <wsdl:port binding="this:ICreditServicesExport1_ICreditServicesHttpBinding" name="ICreditServicesExport1_I
  <soap:address location="http://rt3.in.ibm.com:9078/ExpoCreditScore.modWeb/sca/ICreditServicesExport1"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Figure 1: Edit SOAP address location in WSDL file

- b. Open WSRR Console <http://hostname:port/ServiceRegistry>. Browse to Service Documents -> Load Documents. Select and upload the following documents:
  - i. ExpoCreditScore.mod\_ICreditServicesExport1.wsdl
  - ii. ICreditServices.wsdl
- c. Browse to Business Metadata -> Concepts. Click New and fill in the following details:
  - i. Name: ExpoCheckCreditScore
  - ii. Description: Expo Check Credit Score Service (Third Party)
  - iii. Namespace: <http://scenario3.saas.com/expo>
  - iv. Version: 1.0
- d. Click OK to save the concept.
- e. Browse to Business Metadata -> Concepts -> ExpoCheckCreditScore. Click on Custom Relationships -> New to create a new relationship (check **Error! Reference source not found.**). Fill-in the following details: Name: serviceDocuments. Click Next.
- f. Select the entity type: WSDL Documents. Click Next.
- g. Name: \*Credit\*. Click Next.
- h. Select ExpoCreditScore.mod\_ICreditServicesExport1.wsdl and ICreditServices.wsdl. Click next.
- i. Click Finish to save the Custom Relationship.



**Figure 2: Adding custom relationships to Concept**

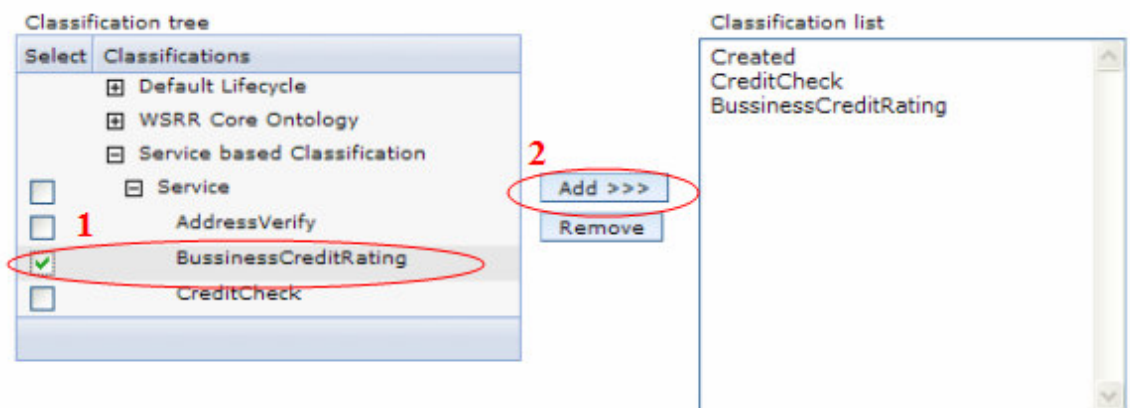
- j. Download and extract the WSRRCustomUI.zip attachment on your local drive.
- k. Browse to WebSphere server profile bin directory and execute the following command to deploy custom UI implementation on WSRR

```
> wsadmin.bat -f wsrrConfiguration.jacl load {PathToExtractedFolder}  
wsrr_config.properties -wsadmin_classpath  
{WSRR_Installation_Dir}/ServiceRegistryClient.jar
```

For more information on deploying custom UI and values in wsrr\_config.properties, please check the following developer works article:

[http://www.ibm.com/developerworks/websphere/library/techarticles/0701\\_smithson4/0701\\_smithson4.html](http://www.ibm.com/developerworks/websphere/library/techarticles/0701_smithson4/0701_smithson4.html)

- l. Restart WSRR server.
2. Add new classifiers for the credit check services
- a. Login to WSRR Console again and change the perspective to User. Browse to Find Service Providers -> All Services.
  - b. Select ExpoCheckCreditScore and click on Classification link on the right side menu.
  - c. Expand Service Based Classification and select BusinessCreditRating. Check Figure 3: Defining classification for ExpoCheckCreditScore concept Figure 3 for more information. Click Add to add it in the Classification List. Click OK to save.



**Figure 3: Defining classification for ExpoCheckCreditScore concept**

### 3. Configure the JK Financials internal mediation proxy layer using WebSphere DataPower SOA Appliance (WDP)

1. Import the WDP configuration containing the routing rules and metering and auditing log actions and transformations:
  - a. Use the “Export/Import Configuration” wizard of the DataPower admin console to import the web services configuration export called export\_DP\_Configurations.zip provided in the Downloads section.
2. Integrate securely with the JK Financials service registry:
  - a. Browse to Objects -> WSRR Server in DataPower admin console. Click on WSRR\_RT3 to open and edit the same.
  - b. Change the SOAP URL for the WSRR server to the one configured in Section 1.  
  
<https://HostName:SSLPortNo/WSRRCoreSDO/services/WSRRCoreSDOPort>
  - c. Click Apply to save the configuration.
  - d. As SSL is enabled in WSRR, the WSRR server certificate needs to be imported in DataPower for successful communication between DataPower and WSRR. Open WSRR server admin console (<http://hostname:port/ibm/console>) and browse to Security -> SSL certificate and key management > Key stores and certificates > NodeDefaultKeyStore > Signer certificates. Check the checkbox in front of dummyclientsigner certificate and click Extract. Provide the filename (with p12 as extension) save click OK to extract the certificate.
  - e. Back in WDP, browse to Objects -> Crypt Certificate in the DataPower admin console and select ValidCret\_WSRR\_RT3 to open and edit.
  - f. Click on the upload button to upload the certificate downloaded in step 2.d above. Check Figure 4 for more information.



Figure 4: Uploading WSRR certificate to DataPower.

- g. Select the certificate just uploaded from the drop down. Provide the password and confirm password as “WebAS” (without quotes). Click Apply to save the configuration.
- h. Browse to Control Panel -> Web Service Proxy. Click on Central\_IT\_Credit\_Score WS Proxy to open and edit it. Click on the status shown under WSDL Status Column (If it is **Error**, as shown in Figure 5). Click Synchronize button in the WSRR Subscription Status pop up window to synchronize the WSDL files between DataPower and WSRR. WSDL Status should change to Synchronizing or Okay. If it is still error, check the log files for details.

WSDL Source Location	Endpoint Handler Summary	WSDL Status	WS-I BP Status	Action
⊕ ExpoCheckCreditScore	1 up / 1 configured	<b>Error</b>		Remove

**Figure 5: Synchronizing WSDL with WSRR Server if WSDL status is Error**

3. Set the correct port for the internal mediation proxy service for the Expo credit check service
  - a. Make sure the port mentioned for Central\_IT\_FS\_Handler (i.e. 9333) is not being used by any other object in DataPower (across domains).
4. Import the public key for the Expo credit check service for message confidentiality:
  - a. Browse Objects -> Crypto certificate -> select ExpoPublicKey.
  - b. Click on the upload button in front of File Name. Select and upload expopubkey.cert provided in the download section. Click Apply to save ExpoPublicKey Crypto Certificate.
5. Add Web Services endpoints for the Usage Logging and Audit Web Services.
  - a. Browse to Control Panel -> Central\_IT\_Credit\_Score\_Proxy. Click on the Policy Tab and select the default response rule, Central\_IT\_Credit\_Score\_Proxy\_default\_reponse-rule.
  - b. Double click on the results asynchronous action to open and edit the configuration, as shown in

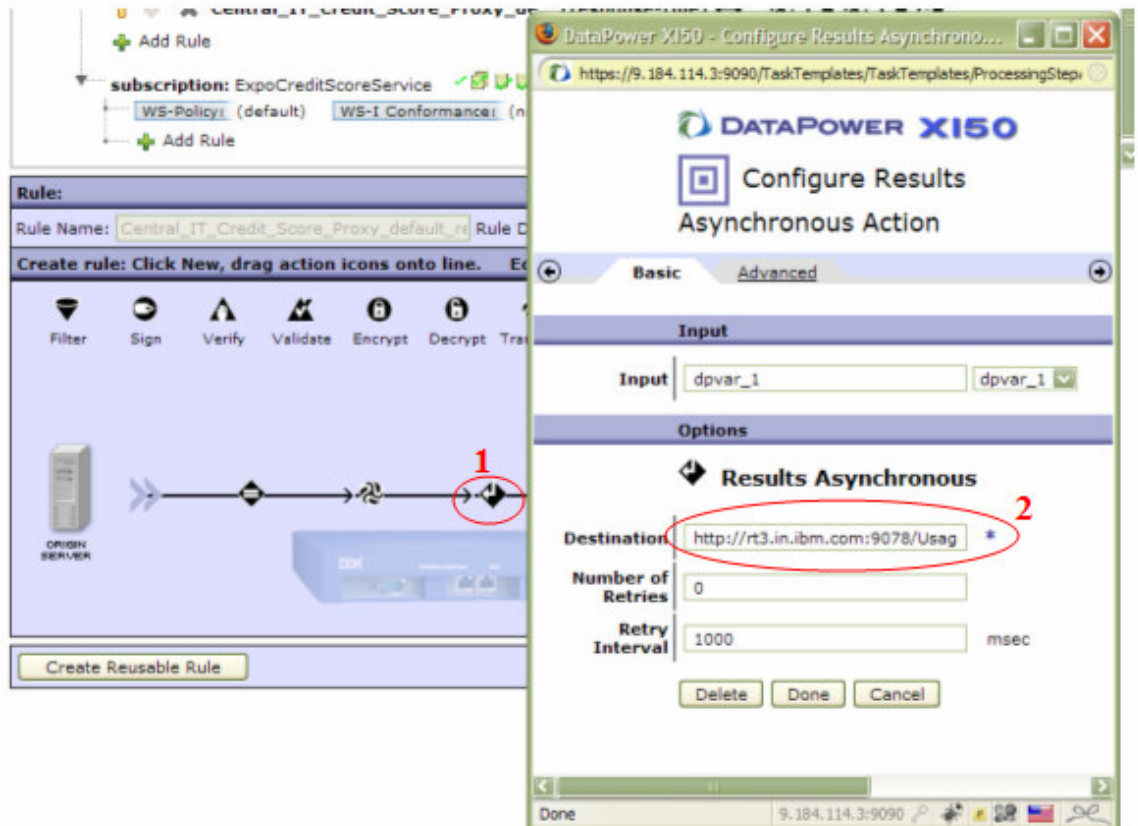


Figure 6 below. Edit the destination field with correct hostname and port no.

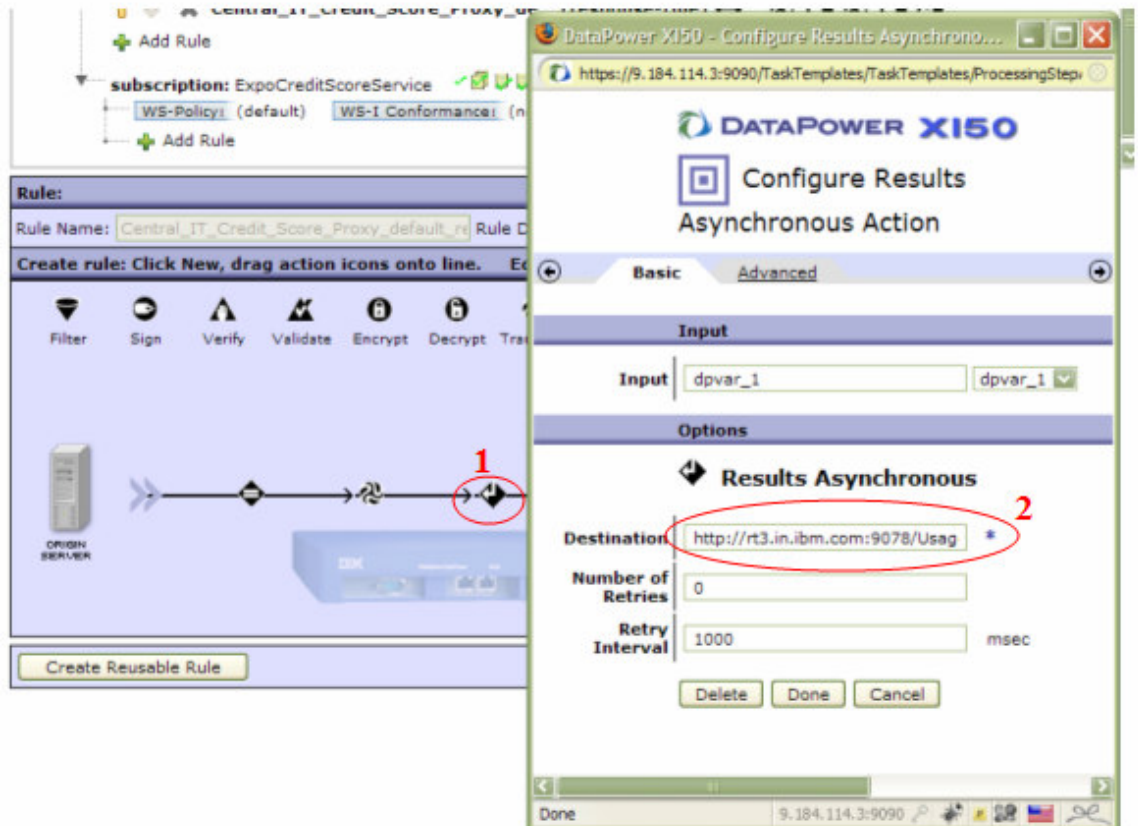


Figure 6: Editing Web Services URL for Usage Log Web Service

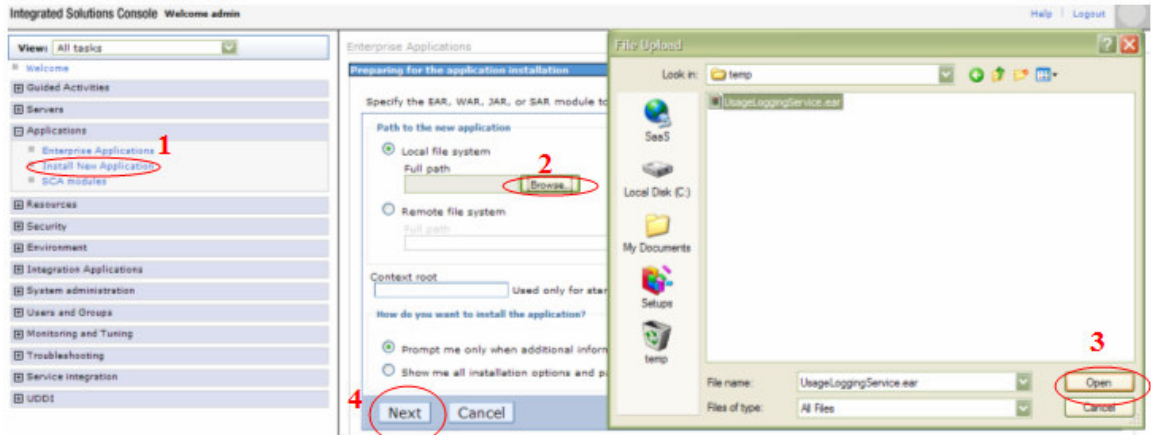
- c. In a similar way, change the hostname and port number in the asynchronous action for Audit Log Service.

#### 4. Configure the Usage Logging and Auditing services in WebSphere Process Server

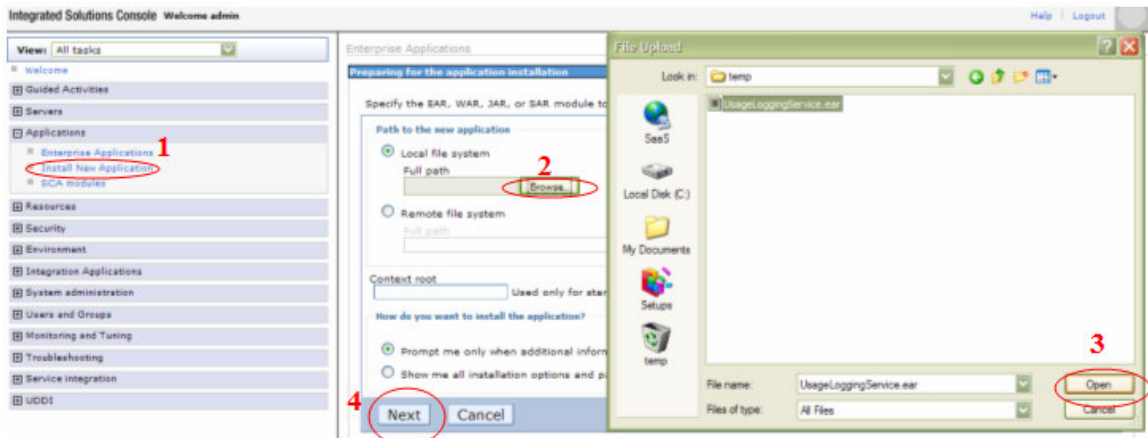
1. Usage Logging Service will make use of a Usage Logging JMX component. For installing and configuring this component, please refer to the deployment instructions associated with the following demo.

<http://www.ibm.com/developerworks/offers/lp/demos/summary/tv-saastuam.html>

2. After configuring the Usage Logging JMX component, deploy the Usage logging service (UsageLoggingService.ear) on WebSphere Process Server 6.1 using the WebSphere Admin Console. Please refer to



for more information.

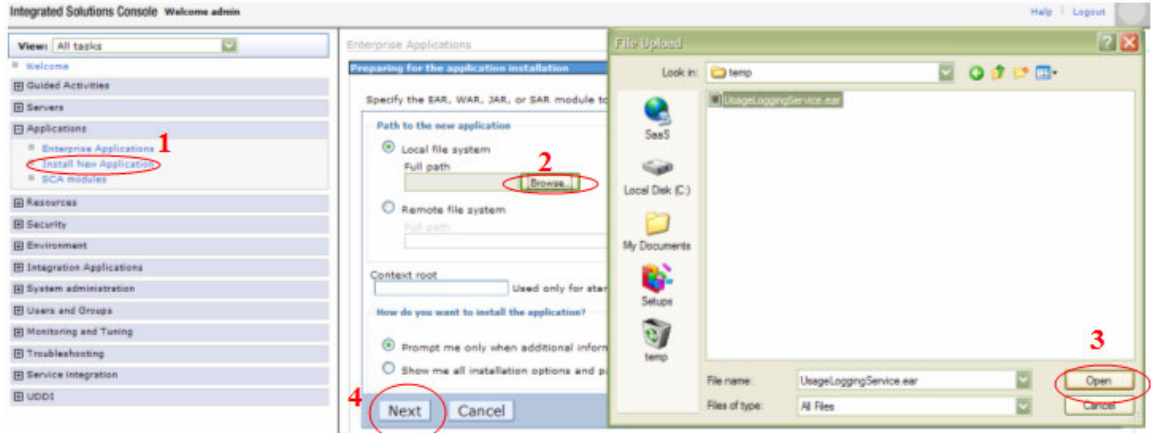


**Figure 7: Deploying Web Services EAR using WebSphere Admin Console**

Pre-requisite:

-> RMI Server is running on the same host where Usage Logging Service is deployed.

3. In the same way, deploy the Auditing service (AuditingService.ear) in WebSphere Process Server 6.1 using the WebSphere Admin Console. Please refer to



for more information.

4. Create the necessary tables required for Auditing Service using AuditService\_SampleSchema.ddl.
5. Create a data source on the same server, where Auditing Service is deployed, with JNDI name jdbc/SRVUSGDB.

#### 4. Configure account codes and rate codes for each department and the Expo credit check service using Tivoli Usage and Accounting Manager (TUAM)

1. Configure the TUAM Universal Data Collector to collect the information from usage log database logged by Usage Logging service. The Universal Data Collector job file, ITUAM\_Job\_File.xml, is provided in the Download section.
2. Configure the rate codes: CSSE1 and CSSE2 for the loan processing and credit card departments respectively
  - a. Login to the admin console of ITUAM using the administration user ID and password. The URL for the administration console on ITUAM server looks like [http://<ituam\\_server\\_name>:11052/ibm/console](http://<ituam_server_name>:11052/ibm/console).
  - b. Navigate to **Usage and Accounting Manager -> Chargeback Maintenance -> Rates** and click **New** as shown in the

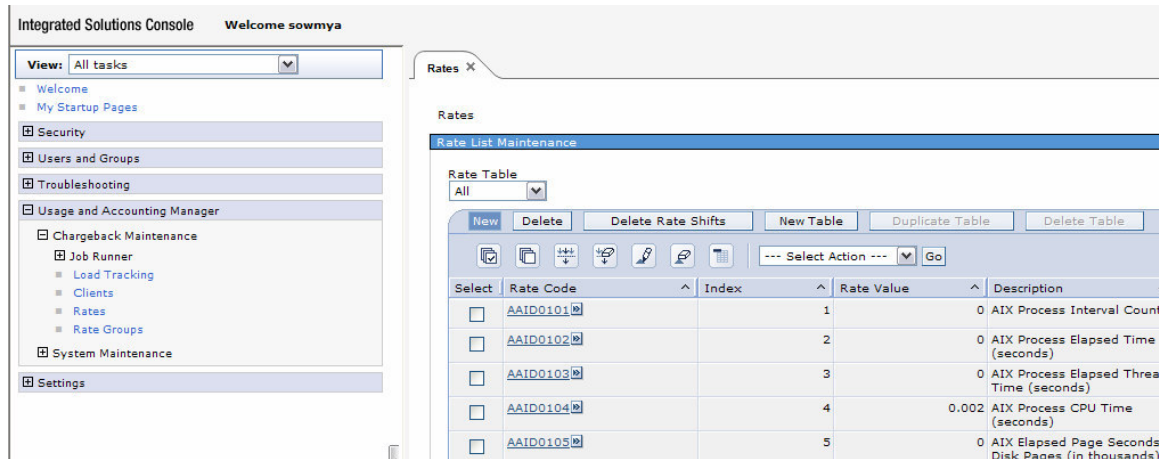


Figure 8 below.

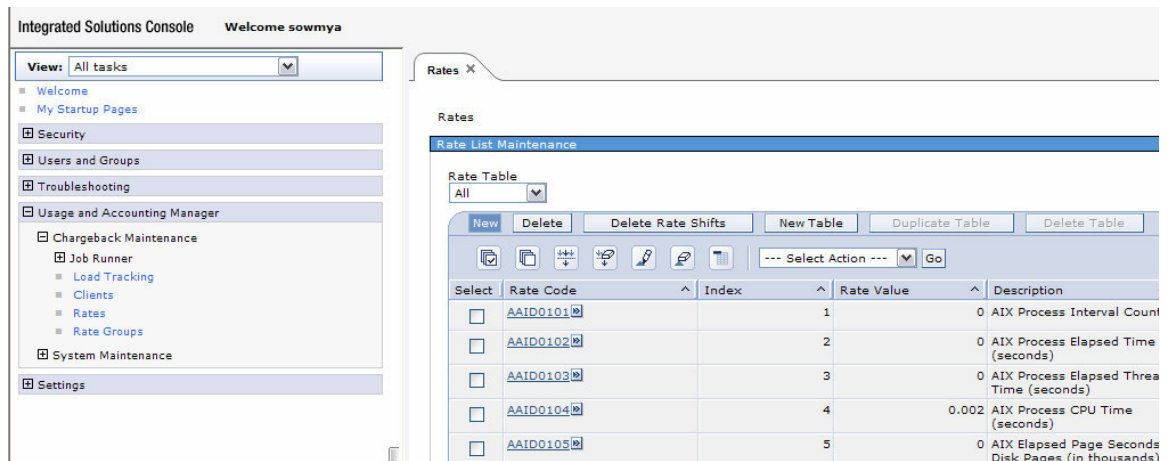


Figure 8: Creating a new rate code using ITUAM admin console

- c. Create a rate code for Expo Credit Check Service usage by Loan Processing Dept.

Enter the following details as shown in the Figure 9 below.

- i. Rate Code -> CSSE1
- ii. Rate Value -> 0.2
- iii. Description -> Expo Credit Check Service Charge for Credit Card Dept.
- iv. Detail Description -> Charge \$2 for 10 invocations.

The screenshot shows the 'Rates Maintenance' window in the ITUAM administration console. The 'General Properties' section is filled with the following information:

- Rate Code:** \*CSSE1
- Description:** Expo Credit Check Service Charge for Credit Card Dept.
- Rate Value:** 0.2
- Rate Index:** 482
- Rate Table:** STANDARD
- Rate Group:** All Unassigned

The 'Detail' section contains several checkboxes and dropdown menus:

- Use 4 decimals for rate
- Rate is per thousand
- Do not adjust for zero cost
- Flat fee money charges
- CPU value
- Resource Conversion:** Default
- Conversion Factor:** 0
- Resource Decimals:** Default
- Report Flag 1:**
- Report Flag 2:**
- Currency Symbol:**

The 'Detail Description' field contains: Charge \$2 for 10 invocations. There is also a 'Comments' field which is currently empty. At the bottom, there are buttons for 'Edit Rate Shifts', 'Delete Rate Shifts', 'OK', and 'Cancel'.

**Figure 9: Creating rate code for Expo Credit Check Service for Credit Card Dept**

- d. In a similar manner, create the other rate code with following details:
  - i. Rate Code -> CSSE2
  - ii. Rate Value -> 0.2
  - iii. Description -> Expo Credit Check Service Charge for Loan Processing Dept.
  - iv. Detail Description -> Charge \$2 for 10 invocations.
3. Configure account codes for each department.

ITUAM uses account codes for identifying entities for billing and reporting for clients. We use an account code to aggregate usage data for tenant specific usage of different types of resources. The following steps describe how to create account codes for each tenant organization using ITUAM administration console.

- a. Using the ITUAM administration console, navigate to **Usage and Accounting Manager -> Chargeback Maintenance -> Clients** and click New.
- b. Create an account code called DPT1 for charging Credit Card Processing Department based on the individual services used by the department. Enter the following fields as shown in the Figure 10 below.
  - i. Account Code -> DPT1
  - ii. Account Name -> Credit Card Processing Department

Clients

Client Maintenance

---

**General Properties**

Account Code  
\*

Account Name

Alternate Account Code

Rate Table  
 ▼

Invoice Contact  
 ▼

Action Codes

**Figure 10: Creating account code for Credit Card Processing Department (DPT1)**

- c. Click on “Budgets” button and add the rate codes by clicking the root element and then selecting “Add Resource Budget”. These rate codes are created in Step 4.2 above. Please have a look at

Client Budget Maintenance

Account Code DPT1 Year 2008

Overall Account Budget  Budget Detail

CSSE1

	Budget Amount	Budget Unit
Total	<input type="text" value="0"/>	<input type="text" value="0"/>
	<input type="button" value="Divide Total"/>	<input type="button" value="Divide Total"/>
	<input type="button" value="Sum Periods"/>	<input type="button" value="Sum Periods"/>
Period 1	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 2	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 3	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 4	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 5	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 6	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 7	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 8	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 9	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 10	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 11	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 12	<input type="text" value="0"/>	<input type="text" value="0"/>

Figure 11 below for more information.

Clients

**Client Budget Maintenance**

Account Code DPT1      Year 2008    Add Year    Delete Year

Overall Account Budget    **Budget Detail**

CSSE1

	Budget Amount	Budget Unit
Total	<input type="text" value="0"/>	<input type="text" value="0"/>
	<input type="button" value="Divide Total"/>	<input type="button" value="Divide Total"/>
	<input type="button" value="Sum Periods"/>	<input type="button" value="Sum Periods"/>
Period 1	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 2	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 3	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 4	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 5	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 6	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 7	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 8	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 9	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 10	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 11	<input type="text" value="0"/>	<input type="text" value="0"/>
Period 12	<input type="text" value="0"/>	<input type="text" value="0"/>

**Figure 11: Rate codes for DPT1 account code.**

- d. In a similar manner, create account code (DPT2) for Loan Processing Department with Rate code (CSSE2).
4. For more details on TUAM configuration steps please see the attached instructions document in this demo:

<http://www.ibm.com/developerworks/offers/lp/demos/summary/tv-saastuam.html>

## 5. Downloads

Sl. No	Attachment Name	Description
1.	ExpoCreditScore.mod_ICreditServicesExport1.wsdl	WSDL document of Expo Credit Score Service.
2.	ICreditServices.wsdl	WSDL document for Expo Credit Score Service referred by ExpoCreditScore.mod_ICreditServicesExport1.wsdl
3.	DataPower_Domain.zip	DataPower domain backup with WS-Proxy configuration

4.	UsageLoggingService.ear	Usage Logging Web Services EAR file.
5.	ExpoCreditCheckService.ear	Expo Credit Score Web Service EAR file.
6.	AuditingService.ear	Audit Log Web Service EAR file.
7.	AuditService_SampleSchema.ddl	Audit Log sample database schema DDL file.
8.	UsageLog_SampleSchema.ddl	Usage Log sample database schema DDL file.
9.	ExpoCreditScore_SampleSchema.ddl	Credit Score sample database schema DDL file.
10.	CreditCardProcessingDeptJobFile.xml	Sample ITUAM XML job file for Credit Card Processing Dept.
11.	LoanProcessingDeptJobFile.xml	Sample ITUAM XML job file for Loan Processing Dept.
12.	WSRRCustomUI.zip	WSRR Custom UI zip file
13.	wrrConfiguration.jacl	WSRR Configuration JAACL script.
14.	wrr_config.properties	WSRR Configuration Properties file.
15.	SourceCode_PI.zip	Project Interchange file containing source code of the artifacts.
16.	expopubkey.cert	Public key certificate of Expo Credit Score Service.
17.	expostore.jks	Java Key Store containing public and private keys for Expo Credit Score Service