
SMART METER

TEXAS

Application Programming Interface (API) Guide

Table of Contents

1.	Executive Summary	6
1.1	Purpose.....	6
1.2	Scope	7
2.	System Overview.....	9
2.1	SMT Integration Overview.....	9
2.2	List of Available SMT Application Programming Interface (API)	11
3.	SMT FTPS (M1) Interfaces	13
3.1	FTPS Interfaces Overview.....	13
3.2	FTPS Server Folder Structure.....	14
3.3	Connecting to FTPS Server.....	15
4.	Common SMT Web Service Interface Definition.....	16
4.1	SOAP Header	16
4.2	SOAP Fault	17
5.	Web Services Interface Security.....	18
5.1	Security Pre-requisite for Interfaces	18
5.2	WS-Security Implementation Features	19
5.3	Trust Relationship Establishment.....	20
5.3.1	Trust Relationships in Production.....	20
5.3.2	Trust Relationships in Test.....	20
5.4	Validation Parameters and Steps	21
5.4.1	Example SOAP Envelope with UserName Token	21
5.4.2	Example Signed SOAP Envelope	22
5.4.3	Validation Parameters	25
5.4.4	Validation Steps	25
6.	SMT Web Service Interfaces.....	26
6.1	SMT Ad-hoc Meter Usage Interface	26
6.1.1	Interface Overview	26
6.1.2	Schema Definitions	29
6.1.3	Schema and Sample Messages	29
6.2	SMT Re-branding Interface.....	36
6.2.1	Interface Overview	36
6.2.2	Schema Definitions	37

6.2.3	Schema and Sample Messages	38
6.3	SMT Third-Party Interfaces.....	45
6.3.1	Interface Overview	45
6.3.2	Schema Definitions	46
6.3.3	Schema and Sample Messages	46
7.	Ad-hoc Usage soapUI Starter Kit	65
7.1	Overview of the soapUI Starter Kit for Ad-hoc Usage Queries	65
7.1.1	Intended Audience.....	65
7.1.2	About soapUI	65
7.1.3	About this Starter Kit	65
7.2	Prerequisites.....	66
7.2.1	Environmental Setup	66
7.2.2	Key Store	67
7.3	Overview of Steps for using the Starter Kit.....	68
7.4	Setting up the Starter Kit.....	69
7.5	SSL Configuration.....	70
7.6	HTTP Header Configuration.....	73
7.7	WS-Security Configuration	76
7.8	Invoking Web Service	82
7.8.1	URLs for Invoking Ad-hoc Usage Queries	82
7.8.2	Staging URLs.....	82
7.8.3	Applying a Web Services Signature to an Outgoing Request	84
7.9	Troubleshooting	88
8.	Appendices	90
8.1	Glossary of Terms	90
8.2	Certification Authorities (CA) List.....	91
8.3	Decrypting the LSE file.....	92

Figures and Tables

Figure 1: System context diagram – M1 Implementation	9
Figure 2: System context diagram – Web Service API highlighted	10
Figure 3: SMT FTPS Components.....	13
Figure 4: Ad-hoc Meter Usage Interface – High Level System View	26
Figure 5: Re-branding Overall Interaction Diagram	37
Figure 6: Re-branding Interaction Operations	38
Table 1: SMT M1 Interfaces	12
Table 2: SMT Re-branding API.....	12
Table 3: SMT Third-Party Agreements API Requests and Responses.....	12
Table 4: SMT Validation Parameters.....	25
Table 5: Ad-hoc Meter Usage Message Volumes	27
Table 6: Ad-hoc Meter Usage Message Limitations.....	28
Table 7: Ad-hoc Meter Usage Request Invocation Pattern.....	29
Table 8: Ad-hoc Meter Usage Request Schema Elements	29
Table 9: Ad-hoc Meter Usage with Uploaded File Request Schema Elements.....	31
Table 10: Ad-hoc Meter Response Schema Elements.....	32
Table 11: Ad-hoc Meter Usage Report File Request Naming Convention	32
Table 12: Ad-hoc Meter Usage Report Status Request Schema Elements	34
Table 13: Ad-hoc Meter Usage Report Status Response Schema Elements	34
Table 14: Re-branding Request Schema Elements	39
Table 15: SMT Re-branding Daily Sync Response Schema Elements	40
Table 16: SMT Re-branding Interval Sync Response Schema Elements	41
Table 17: SMT Re-branding Monthly Sync Response Schema Elements	43
Table 20: Third-Party Energy Data Agreement Request Schema Elements	47
Table 21: Third-Party Energy Data Agreement Request Acknowledgement Schema Elements.....	49
Table 22: Third-Party Energy Data Agreement Extension Request Schema Elements.....	50
Table 23: Third-Party Energy Data Agreement Extension Request Acknowledgement Schema Elements	51
Table 24: Third-Party In-Home Device Services Agreement Request Schema Elements.....	53
Table 25: Third-Party In-Home Device Services Agreement Request Acknowledgement Schema Elements	55
Table 26: Third-Party In-Home Device Agreement Request Schema Elements	56
Table 27: Third-Party In-Home Device Agreement Request Acknowledgement Schema Elements	60
Table 28: Third-Party Re-Add In-Home Device Request Schema Elements	60
Table 29: Third-Party In-Home Device Agreement Request Acknowledgement Schema Elements	62

Table 30: Third-Party Scheduled Usage Report Request Schema Elements	63
Table 31: Third-Party Scheduled Usage Report Request Acknowledgement Schema Elements.....	64
Table 32: WS-Security Configuration Signature Parts Information.....	81
Table 33: Glossary of Terms.....	90

1. Executive Summary

1.1 Purpose

The Smart Meter Texas (SMT) Application Programming Interface (API) Guide is the primary document covering the implementation of automated data integration with SMT for the Texas market participants with existing Smart Meter programs and services in the Texas territories covered by American Electric Power (AEP), CenterPoint Energy (CNP), Texas New Mexico Power (TNMP) and Oncor Electric Delivery.

The API Guide is a consolidation of artifacts developed for the Texas market participants, with the objective that this document be the main source of information regarding the system integration via APIs with the SMT. This API Guide is a “living” document, continually evolving based on upgrades to the SMT functionality driven by national standards as they are developed, developments in Advanced Metering System (AMS) technology, and market input.

This document aims to help the Transmission & Distribution Service Providers (TDSP), Retail Electric Providers (REP) and Third-Party Service Providers (Third-Party) to gain better understanding of the functionality available to the TDSPs, REPs and Third-Parties to integrate their AMS systems with SMT via API and how to use these APIs.

The API Guide explains in detail the technical processes to invoke the file based transfer and the web service interfaces and the data available through these APIs. This document provides some examples of the Extensible Markup Languages (XML) used by the web service interfaces, and the soapUI Starter Kit section of the API Guide outlines the process for the TDSPs, REPS and Third-Parties use for invoking the SMT Ad-Hoc Query Web Service interfaces. This document also addresses associated authorization, validation and security issues.

The API Guide is only available to all registered TDSP, REP and Third-Party users of the Smart Meter Texas website. The target audiences of this document are assumed to have basic technical expertise in these areas:

- Downloading applications from the Internet
- Installation and setup of applications
- Creation and editing of XML documents
- Internet security including user ids, passwords, and the use of X.500 certificates
- Network communications including URLs, IP addresses and common network troubleshooting commands such as ping, telnet, FTP, etc.

1.2 Scope

The API Guide contains the following sections related to the SMT interfaces available for integration between the AMS system of the Texas market participants and SMT. This document does not cover In-Home Device APIs that are available in SMT. For information of the In-Home Device APIs, please refer to the SMT In-Home Device Guide.

The following is a quick reference to locate the information that the Texas market participants may be looking for:

API Topics	Interested Party	Sections
AMS System Integration Overview	Business Analysts	Section 2
SMT FTPS Interfaces	Technical Analysts, Business Analysts	Section 3
SMT Web Service Interfaces	Technical Analysts, Business Analysts	Section 6
Examples of Extensible Markup Language (XML) for Web Service interfaces	Technical Analysts, Business Analysts	Section 6
How to Integrate with SMT for Ad-Hoc Queries (soapUI Starter Kit)	DataPower / API Technical Integration Team	Section 6.3
Security Requirements to Interface with SMT	DataPower / API Technical Integration Team	Section 5
Glossary of Terms	All Market Participants	Section 8

Section 3: SMT FTPS Interfaces

This section covers the file-based interfaces using Secure File Transfer Protocol (FTPS) implemented during the first phase (M1) of the SMT portal. These APIs enable:

- Meter usage data (Interval reads, register reads and monthly billed usage data)
- Master data (non-usage data including meter attributes, premise info and REP of record)

Section 5: SMT Interface Security

This section describes Smart Meter Texas (SMT) interface security requirements necessary for integration via API and the API Starter Kit configuration set-up. It includes technical details required on API requests from the TDSP (i.e. user credential in SAML token that is part of SOAP header), and API requests from REPs and Third-Parties (i.e. user credential passed in a UserName Token) that is part of SOAP header. It also includes SMT requirements for CA issued certificates (SSL and Signer Cert) in its production environment.

Section 6: SMT Web Service Interfaces

This section lists and explains all the web service APIs that are available in SMT. Examples of the APIs are provided. The following is the list of all SMT web service APIs:

- Ad-hoc Meter Usage Query
- Re-branding
- On Demand Read

Section 6.3: soapUI Starter Kit for the Ad-Hoc Query Usage Data Interface

The soapUI Starter Kit is a tool that was developed by SMT to help the REPs and Third-Parties to invoke the SMT Ad-Hoc Query Usage Data web service interface.

The section explains the prerequisites and steps to use the soapUI Starter Kit for the Ad-Hoc Query Usage Data web service interface. It includes the UAT/Staging SSL certificates required by the interfaces.

In addition, the section also describes the process for how the REPs and Third-Parties can request the Webservice description language (WSDL) files. After the requesters have integrated with the SMT using the Starter Kit, the WSDL files for the related services will be provided by SMT upon request. These files can be utilized to build client applications to consume the SMT services for Ad-Hoc query of Usage Data.

Section 8: Appendices

This section includes the glossary of terms, the list of Certificate Authorities acceptable to SMT, and the decryption steps of the LSE files.

2. System Overview

2.1 SMT Integration Overview

Figure 1 represents the solution as a single object and identifies the logical interfaces between the SMT Web Portal and Data Repository and external entities implemented during M1 phase. It shows the information and control flows that cross the system boundary. The Integration System Context is focused on Meter master data & usage data end-to-end functionality.

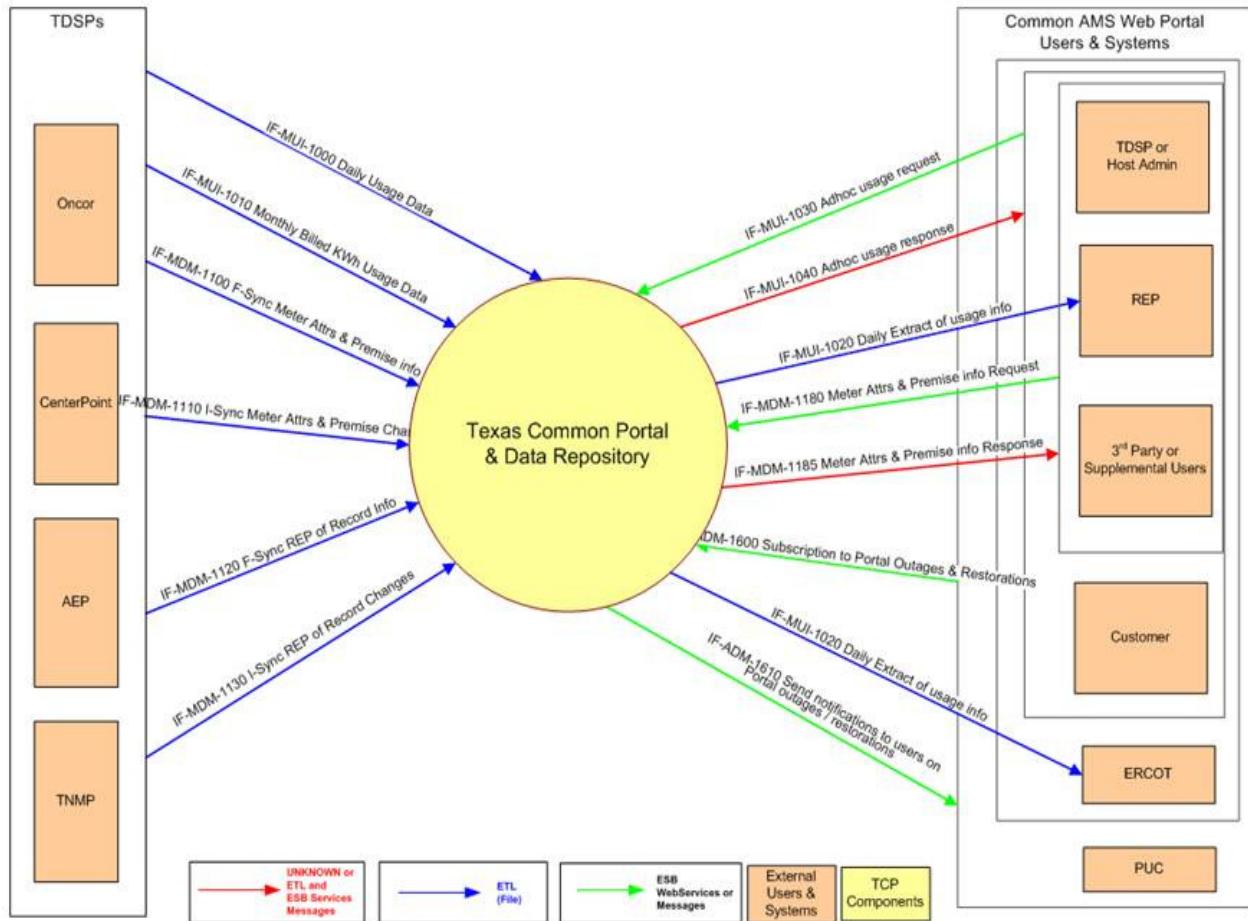


Figure 1: System context diagram – M1 Implementation

Figure 2 presents the system context diagram for SMT with the 2 web service API highlighted.

- Ad-hoc Meter Usage Query
- Re-branding functionality

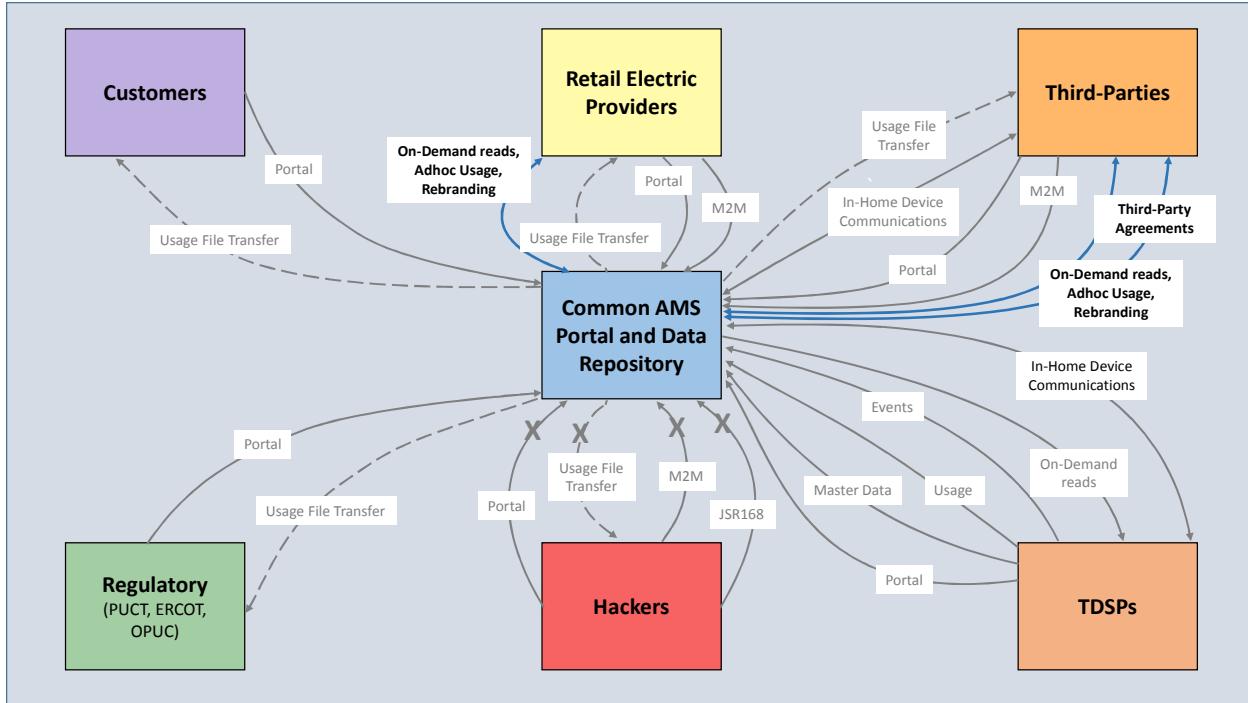


Figure 2: System context diagram – Web Service API highlighted

SMT will receive ad-hoc meter usage query from the TDSPs, REPs and Third-Parties systems. SMT will also receive re-branding requests from REPs and Third-Parties' systems. SMT performs validations as described in this document, and returns the API data to the requester that made the requests.

The interfaces described in this document have been defined based on Service Oriented Architecture (SOA) principles. The interfaces have been specified in the Web Service Description Language (WSDL). Web Services between requesters and SMT will be secured using mutual authentication implemented with Secure Socket Layer (SSL).

2.2 List of Available SMT Application Programming Interface (API)

Based on the review of the AMIT requirements, approximately 13 different logical interfaces were in scope for the Common AMS Web Portal Release 1 to provide the M1 functionality addressing the following subject areas

- Meter usage data (Interval reads, register reads and monthly billed usage data)
- Master data (non-usage data including meter attributes, premise info and REP of record)

Table 1 lists the logical interfaces identified for Release 1 addressing M1 functionality and discusses high-level characteristics of each interface. It also indicates if data related to each interface must be migrated into the portal when the portal transitions to production. The exposition here is at high-level and the next section of this document covers these listed interfaces in detail.

Interface Number	Interface Description	Source	Destination	Interface Type (File, DB, Messaging, RPC, etc.)	Real time / Batch
IF-MUI-1000	Send meter usage data with 15 minutes interval and daily register read	TDSPs	Smart Meter Texas Portal	File	Batch
IF-MUI-1010	Send meter monthly billed usage data	TDSPs	Smart Meter Texas Portal	File	Batch
IF-MUI-1020	Send daily extract of usage info since the last extract for all usage data of the REP of Record's ESIIDs	Smart Meter Texas Portal	REPs	File	Batch
IF-MUI-1030	Send ad-hoc queries/request to get meter usage data with specific date range, usage data interval and one or multiple ESIIDs	Third-Parties REPs TDSPs Customers	Smart Meter Texas Portal	Web service	Real-time
IF-MUI-1040	Meter usage data response to ad-hoc queries for specific date range, usage data interval and one or multiple ESIIDs	Smart Meter Texas Portal	Third-Parties REPs TDSPs Customers	Web service / File	Real-time /Batch
IF-MDM-1100	Full Sync of Meter attributes (Non-Usage Data)	TDSPs	Smart Meter Texas Portal	File	Batch
IF-MDM-1110	Incremental Sync of meter attributes (Non-Usage Data) changes	TDSPs	Smart Meter Texas Portal	File	Batch
IF-MDM-1120	Full Sync of REP of Record	TDSPs	Smart Meter Texas Portal	File	Batch
IF-MDM-1130	Incremental Sync of REP of Record changes to update data repository with the relationship (new, switch or move out)	TDSPs	Smart Meter Texas Portal	File	Batch
IF-MDM-1140	Full Sync of Premise Info	TDSPs	Smart Meter	Web service	Real-time

Interface Number	Interface Description	Source	Destination	Interface Type (File, DB, Messaging, RPC, etc.)	Real time / Batch
			Texas Portal	/ File	/ Batch
IF-MDM-1150	Incremental Sync of Premise Info	TDSPs	Smart Meter Texas Portal	Web service / File	Real-time / Batch

Table 1: SMT M1 Interfaces

The Re-branding API is being made available for TDSPs, REPs and Third-Parties in a subsequent release. Table 2 shows the set of the Re-branding API requests and responses.

Interface Request Message	Response	Invocation Type
Re-Branding Request	IntervalSyncResponse	Synchronous
	DailySyncResponse	Synchronous
	MonthlySyncResponse	Synchronous
	SOAPFault	Synchronous

Table 2: SMT Re-branding API

The following Third-Party Agreement APIs are being made available for REPs and Third-Parties in Release 4.3 of SMT. Table 3 shows the set of Third-Party Agreements API requests and responses.

Interface Request Message	Response	Invocation Type
SMTxPEnergyDataAgreementRequest	SMTxPEnergyDataAgreementAck	Synchronous
	SOAPFault	Synchronous
SMTxPEnergyDataAgreementExtRequest	SMTxPEnergyDataAgreementExtAck	Synchronous
	SOAPFault	Synchronous
SMTxPHANServicesAgreementRequest	SMTxPHANServicesAgreementAck	Synchronous
	SOAPFault	Synchronous
SMTxPHANDeviceAgreementRequest	SMTxPHANDeviceAgreementAck	Synchronous
	SOAPFault	Synchronous
SMTxPHANDeviceReAddRequest	SMTxPHANDeviceReAddAck	Synchronous
	SOAPFault	Synchronous
SMTxPScheduledReportRequest	SMTxPScheduledReportAck	Synchronous
	SOAPFault	Synchronous

Table 3: SMT Third-Party Agreements API Requests and Responses

3. SMT FTPS Interfaces

3.1 FTPS Interfaces Overview

Figure 3 depicts the components and processing related to the SMT secured File Transfer Protocol (FTPS) interface available to REPs and Third-Parties.

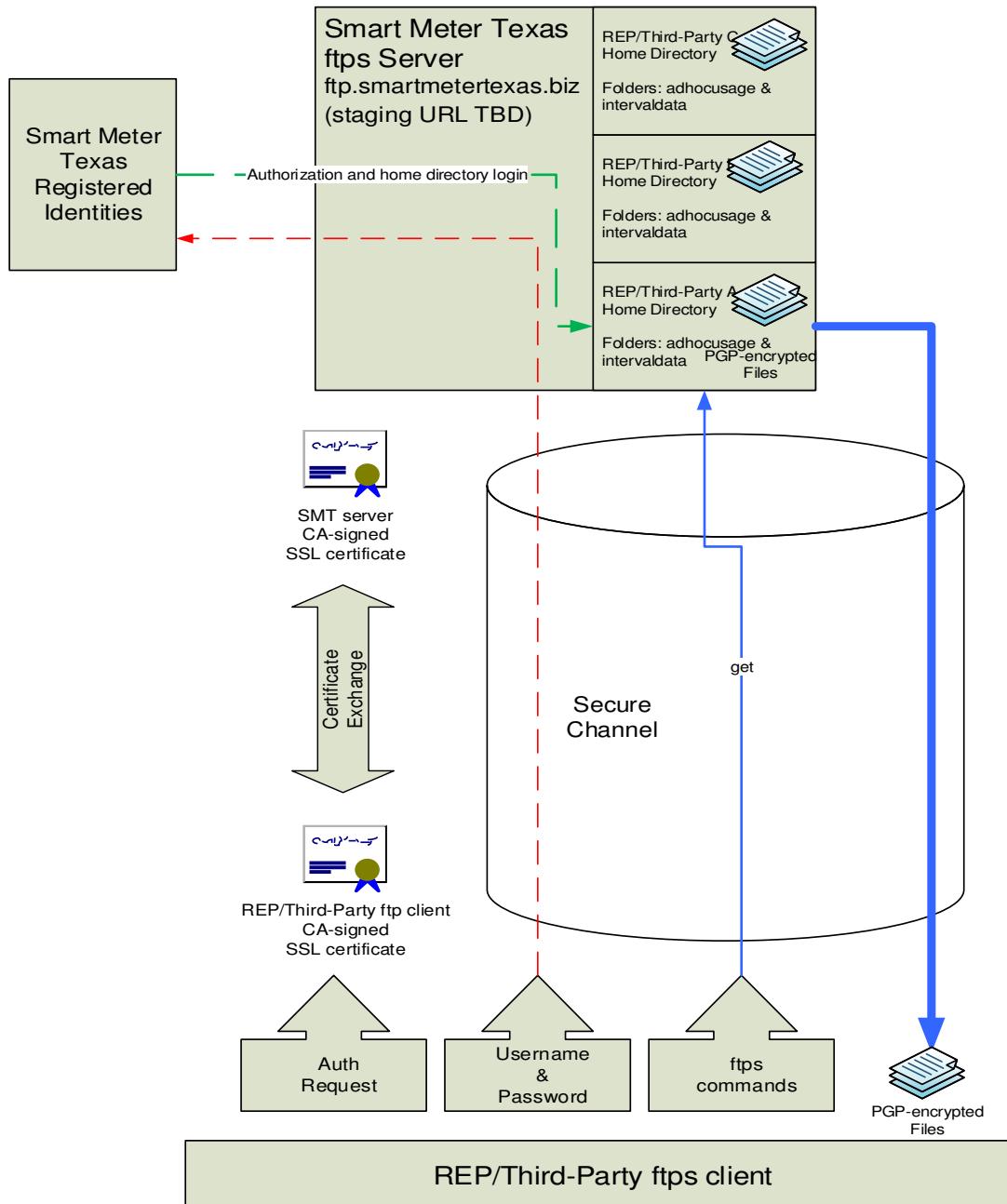


Figure 3: SMT FTPS Components

As depicted, REPs and Third-Parties need several pre-requisites to pull usage data from the Portal:

- Registered and active portal account, including an FTPS username and password
- PGP encryption key pair (public key to Smart Meter Texas Portal)
- An ftps “client” software (commercial and free versions available)
- CA-signed SSL certificate
- Outbound access to port 21

The completion of portal signup results in the establishment of an account, which includes a valid username and password. The registration also results in the creation of a home directory, from which the requester accesses their usage files. It should be noted that the FTPS account will be created by a Smart Meter Texas Portal administrator and is verbally conveyed to the requester's SMT administrator. This account is distinct from the Portal logon account username and password.

At the time of sign-up, the REP or Third-Party must generate a set of PGP encryption keys and provide the public key to Smart Meter Texas Portal. The Portal hosting team will use that public key to encrypt the usage files so that they can only be decrypted using the secret private key.

The REP or Third-Party must obtain a FTPS (FTP over SSL) client software to interact with the FTPS interface. This FTPS software, not to be confused with FTPS software, is needed to request the usage files. These FTPS client software packages are available commercially and in free forms.

Finally, the REP or Third-Party must obtain an SSL certificate signed by a recognized Certificate Authority, such as VERISIGN. These certificates facilitate the mutual authentication of the FTPS client computer and the Smart Meter Texas Portal FTP host computer, and set up a secure channel between those devices encrypted via SSL.

3.2 FTPS Server Folder Structure

FTP server contains two folders. FTPS Sites # 1 for posting the Daily LSE files received from the TDSPs and # 2 for posting the usage response files generated for the Ad-hoc usage requests received from the requesters via SMT Portal or API call.

1. Folder Name – “intervaldata” → Contains the LSE files SMT received from the TDSPs .
 - Example file: 957877905IntervalData20100122123001980.lse.002.799530915
2. Folder Name – “adhocusage” → Contains the SMT generated CSV/LSE files for the ad-hoc usage requests
 - Example files:
IntervalMeterUsagecf024ab76ee195c3c326fe9.CSV.8286294561000
IntervalMeterUsagecf024ab76ee195c3c326fe9.lse.8286294561000
DailyMeterUsage00122c501ff160ca73ad74a7.CSV.799530915

3.3 Connecting to FTPS Server

The FTPS interaction follows a basic sequence depicted in the graphic and described in the text below.

- 1) The REP or Third-Party's FTP client contacts the Smart Meter Texas Portal FTPS server (ftp.smartmetertexas.biz) and requests mutual authentication. This results in the exchange of SSL certificates and the establishment of a secure encrypted channel between the FTPS client device and the Smart Meter Texas Portal FTPS server device. It is important to note that the REP or Third-Party's firewalls must be open to outbound traffic on port 21.
- 2) The REP or Third-Party provides FTPS username as password via the FTPS client software.
- 3) The Smart Meter Texas Portal FTPS server validates that username and password against the Identity database, and establishes user access to the REP or Third-Party's home directory. It should be understood that a REP or Third-Party can only access their own directory and have NO visibility to others' home directories or data.
- 4) Following successful validation of FTPS username and password credentials, the REP or Third-Party may request download of usage files & delete the same using standard FTPS commands. The specific commands should be referenced within the documentation for the chosen FTPS client software package. The Smart Meter Texas Portal FTPS server delivers the usage files to the requester in an encrypted form that can only be decrypted using the REP or Third-Party's private PGP key, through an encrypted tunnel.
- 5) The Smart Meter Texas Portal FTPS server retains the files on the FTPS server for 10 calendar days and will be deleted thereafter.

4. Common SMT Web Service Interface Definition

4.1 SOAP Header

SMT messages use a common security header. Here is an example of a soap header in the request messages:

```
<soapenv:Header>
  <wsse:Security xmlns:wsse=
    "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
    <wsse:UsernameToken wsu:Id="UsernameToken-45"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
      1.0.xsd">
      <wsse:Username>direct_energy</wsse:Username>
      <wsse:Password
        Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
        1.0#PasswordText">
        SMT1234
      </wsse:Password>
      <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-
      message-security-1.0#Base64Binary">NU8/c5iYlhDKzo1Vm5sPig==</wsse:Nonce>
      <wsu:Created>2010-06-10T17:18:26.500Z</wsu:Created>
    </wsse:UsernameToken>
    <wsu:Timestamp wsu:Id="Timestamp-44" xmlns:wsu="http://docs.oasis-
      open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"><wsu:Created>2010-06-
      10T17:18:26.500Z</wsu:Created>
    <wsu:Expires>2010-06-10T17:19:26.500Z</wsu:Expires>
  </wsse:Timestamp>
  </wsse:Security>
</soapenv:Header>
```

The security validation using the information in the soap Header will be explained in detail in the next section of this document.

Note that examples of web service messages found in subsequent sections of this document will use an empty soap Header in lieu of this.

4.2 SOAP Fault

SMT uses a common soap Fault for all web service messages. Here is an example of a soap Fault message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"  
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
    instance">  
    <soapenv:Header/>  
    <soapenv:Body>  
        <soapenv:Fault xmlns:m="http://AMSLib">  
            <faultcode>m:SOAPFault</faultcode>  
            <faultstring>1010</faultstring>  
            <detail encodingStyle="">  
                <me:SOAPFault_element xmlns:me="http://schemas.esb.ams.com/meterusagesource">  
                    <faultcode>1010</faultcode>  
                    <faultstring>Data cannot be loaded because a mandatory data element is missing</faultstring>  
                    <faulttime>12/01/2009 03:53:02 EST</faulttime>  
                    <faultdetails>  
                        <errorcode>userId</errorcode>  
                        <errormessage>CWLBN1009E: The value "" with length 0 must have at least length  
1</errormessage>  
                    <applicationname>Ad-HocMeterUsage</applicationname>  
                    <modulename>BIM_MeterUsage</modulename>  
                    <srcsystem>Poratl</srcsystem>  
                    <tgtsystem>Oracle/ETL</tgtsystem>  
                </faultdetails>  
            </me:SOAPFault_element>  
        </detail>  
    </soapenv:Fault>  
    </soapenv:Body>  
</soapenv:Envelope>
```

5. Web Services Interface Security

This section describes security requirements necessary for all the web services interfaces supported by the Smart Meter Texas (SMT).

5.1 Security Pre-requisite for Interfaces

This section lists interface security pre-requisites:

- All communication to SMT from REPs, Third-Parties and TDSPs will be over 2-Way Secure Sockets Layer (SSL). For each TDSP, a VPN channel has to be established between SMT and TDSP systems. In the case of TDSPs, SSL will be employed over a virtual private network (VPN).
- For API requests from TDSPs, SMT requires that a user credential be passed in a SAML token that is part of SOAP header. The specific of SAML token is described in a later section of this document.
- For API requests from REPs and Third-Parties, SMT requires that a user credential be passed in a UserName Token that is part of SOAP header. SMT does not require password to be passed as part of UsernameToken. The specifics of UsernameToken are described in a later section of this document.
- SMT will only accept CA issued certificates (SSL and Signer Cert) in its production environment. Self-signed certificates (SSL or Code Signed) will not be accepted.
- SMT will accept self-signed certificates for testing purposes only. Although the self-signed certificates will be accepted for testing, CA certificates are preferred.
- It is assumed that partners will use Class 3 certificates.
- The system account for TDSPs, REPs and Third-Parties should exist in the SMT user repository and be known to TDSP, REP and the Third-Party in advance. The system account is case sensitive.
- SMT will only accept signed requests as per WS-Sec 1.1 specification for all the incoming requests. The SMT signature processing module does explicit checks for the signature of UserNameToken, Simple Object Access Protocol (SOAP) Body and Time stamp. Although the SOAP request may be signed, the request will be reject if these elements are not signed.
- Signature Confirmation is not implemented.

5.2 WS-Security Implementation Features

This section describes the implementation of WS-Security by SMT.

- SMT is going to have a Certificate Management System in place. Hence it will use certificates in its certificate store for signature verification purposes. If a certificate is passed as part of the SOAP envelope, the SML signature processing module ignores it.
- SMT does certificate chain validation before using certificates for Signature validation.
- A SOAP Fault is thrown with an appropriate Fault Code in the event of error.
- The rule processing stops at the action which results into error.
- SMT receives entity information from HTTP header variable named ENTITY_NAME. The value of the header variable should be the REP or Third-Party's Company name or the string that uniquely identifies the Company.

5.3 Trust Relationship Establishment

Entities are required to establish a trust relationship with the Third-Parties, REPs and TDSPs. Entity creation is an out-of-band process.

5.3.1 Trust Relationships in Production

These entities are required by SMT:

- Signer Certificate – There should be only one CA-issued Signer Certificate that should be given to SMT.
- Value of HTTP header variable ENTITY_NAME.
- SSL Intermediate CA certificate.

This entity is provided by SMT to a REP or Third-Party:

- SystemAccount – There will be one SystemAccount that will be created for each Third-Party or REP. The System Account is used to validate the user against the SMT account database. The System Account corresponding to the Third-Party or REP will be provided.

5.3.2 Trust Relationships in Test

These entities are required by SMT:

- Signer Certificate - Self Signed Certificate or CA certificate. The number of this certificate is 1.
- Value of HTTP header variable ENTITY_NAME.
- SSL Certificate – SSL Certificate should be provided to SMT when a Third-Party or REP is using a self-signed certificate.
- Intermediate CA Certificate – This may be required if a CA certificate is going to be used.

Entity provided by SMT to the REP or Third-Party:

- SystemAccount – There will be one SystemAccount that will be created for each Third-Party or REP. The SystemAccount is used to validate the user against the SMT account database. The SystemAccount corresponding to the Third-Party or REP will be provided.

5.4 Validation Parameters and Steps

5.4.1 Example SOAP Envelope with UserName Token

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  xmlns:smt="http://schemas.esb.ams.com/smtxpmessaging">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
      secext-1.0.xsd">
      <wsse:UsernameToken>
        <wsse:Username>CNP</wsse:Username>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <smt:processPricingMessage>
      <SMTxPPriceSignalRequest>
        <RequesterType>0</RequesterType>
        <RequesterAuthenticationID>123456789</RequesterAuthenticationID>
        <RequesterID>REPAdmin1</RequesterID>
        <RequestPriority>L</RequestPriority>
        <AddressBlock>
          <AddressList>
            <!--0 to 10000 repetitions:-->
            <Address>
              <ESIID>1234567890123456789012</ESIID>
              <MeterSerialNumber>60333050</MeterSerialNumber>
              <DeviceMACAddr>PRC-MACADDR1</DeviceMACAddr>
            </Address>
          </AddressList>
        </AddressBlock>
        <PriceMessageBlock>
          <ProviderID>123456789</ProviderID>
          <RateLabel>Rate Label1</RateLabel>
          <IssuerEventID>7001</IssuerEventID>
          <CurrentTime>2009-12-14T16:30:00</CurrentTime>
          <UOM>1</UOM>
          <Currency>USD</Currency>
          <PriceTier>1</PriceTier>
          <PriceTrailingDigit>3</PriceTrailingDigit>
        </PriceMessageBlock>
      </SMTxPPriceSignalRequest>
    </smt:processPricingMessage>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<RegisterTier>1</RegisterTier>
<StartTime>2009-12-18T16:30:00</StartTime>
<Duration>55</Duration>
<Price>12777</Price>
<PriceRatio>105</PriceRatio>
<GenerationPrice>12111</GenerationPrice>
<GenerationRatio>95</GenerationRatio>
<AlternateCostDelivered>1111</AlternateCostDelivered>
<AlternateCostUnit>1</AlternateCostUnit>
<AlternateCostTrailingDigit>2</AlternateCostTrailingDigit>
</PriceMessageBlock>
</SMTxPPriceSignalRequest>
</smt:processPricingMessage>
</soapenv:Body>
</soapenv:Envelope>
```

- SMT accepts a UserNameToken from the REP or Third-Party with the UserName element. A password is not required.
- The Username element within UserNameToken should be the System Account assigned to the requester. The System Account is set up after the requester's first admin is registered.
- The RequesterType element should be '3' for Third-Parties.
- The RequesterAuthenticationId should be the DUNS number for the REP or Third-Party.

5.4.2 Example Signed SOAP Envelope

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:smt="http://schemas.esb.ams.com/smtxpmessaging">
  <soapenv:Header>
    <wsse:Security soapenv:mustUnderstand="1"
      xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wsu:Timestamp wsu:Id="Timestamp-6f087de8-475a-4a73-a156-7550dfdb227a"
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <wsu:Created>2010-01-19T21:33:16Z</wsu:Created>
        <wsu:Expires>2010-01-19T21:38:16Z</wsu:Expires>
      </wsu:Timestamp>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
          <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
```

```
<Reference URI="#Id-a73451ca-350b-4ab0-8fb8-4c8a845676fe">
<Transforms>
<Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
</Transforms>
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
<DigestValue>IIADgpAo5apSEJbNFeq84eyK4p4=</DigestValue>
</Reference>
<Reference URI="#Id-7bae2f18-a07b-4602-bc3d-237765fd72d0">
<Transforms>
<Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
</Transforms>
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
<DigestValue>Dw97YWmyoFZ1zEHdfu6yR9dY7zs=</DigestValue>
</Reference>
<Reference URI="#Timestamp-6f087de8-475a-4a73-a156-7550dfdb227a">
<Transforms>
<Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
</Transforms>
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
<DigestValue>ol1HC0SNO3IZldrEaOSxGUFzFn4=</DigestValue>
</Reference>
</SignedInfo>
<SignatureValue>gJbQeqjQYVwcLjTRj30y1u8xEhnC1bPeNUL1Ky83rl6f+ZVjmuBBnAyJ3SHLsD
xi/DXfcKCbsPagSGAxcxch0/HPP+yT6bdIMiLWd/lkGSipBETfWWkeWfAKBLoiQCorbE+j2FxNPKV
R/mU1nrg2iVV9cLM8Bwy2Mu90U969r3c=</SignatureValue>
<KeyInfo>
<wsse:SecurityTokenReference>
<dsig:X509Data xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" >
<dsig:X509IssuerSerial>
<dsig:X509IssuerName>CN=www.REP.com, OU=Retail, O=REP Energy, L=Dallas,
ST=TX, C=US</dsig:X509IssuerName>
<dsig:X509SerialNumber>1262968678</dsig:X509SerialNumber>
</dsig:X509IssuerSerial>
</dsig:X509Data>
</wsse:SecurityTokenReference>
</KeyInfo>
</Signature>
<wsse:UsernameToken wsu:id="Id-a73451ca-350b-4ab0-8fb8-4c8a845676fe"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<wsse:Username>SYSTEMACCOUNT</wsse:Username>
</wsse:UsernameToken>
</wsse:Security>
```

```
</soapenv:Header>
<soapenv:Body wsu:Id="Id-7bae2f18-a07b-4602-bc3d-237765fd72d0"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <smt:processSimpleMessaging>
    <SMTxPSimpleMessageRequest>
      <RequesterType>0</RequesterType>
      <RequesterAuthenticationID>123456789</RequesterAuthenticationID>
      <RequesterID>STAGEREPFIRSTADMIN16JAN</RequesterID>
      <RequestPriority>M</RequestPriority>
      <CallbackUri />
      <AddressBlock>
        <GroupID>799530915</GroupID>
      </AddressBlock>
      <SimpleMessageBlock>
        <MessageID>40004</MessageID>
        <StartTime>2010-01-19T16:39:00</StartTime>
        <DurationTime>5</DurationTime>
        <Message>Simple Message on 2010-01-19</Message>
        <MCTransmission>0</MCTransmission>
        <MCPriority>0</MCPriority>
        <MCConfirmation>0</MCConfirmation>
      </SimpleMessageBlock>
    </SMTxPSimpleMessageRequest>
  </smt:processSimpleMessaging>
</soapenv:Body>
</soapenv:Envelope>
```

- SMT accepts signed message from its partner. In the event that it receives an unsigned message or message with invalid signature, the request will be rejected and SOAP Fault will be sent.
- For signed messages, SMT mandates that the SOAP Body, UsernameToken and TimeStamp elements be explicitly signed as in the example above. If anyone of the elements is not signed, the request will be rejected and SOAP Fault will be sent.
- The Token reference type is IssuerSerial. SMT will not extract a certificate from the SOAP Header for Signature validation. It uses certificates from its certificate store for Signature Validation. Therefore, it is imperative that certificates are exchanged prior to commencement of transaction.

5.4.3 Validation Parameters

Following table describes validation parameters, source and purpose:

Parameter	Source	Purpose
Subject Name	SSL Client Certificate	To know the originator of request
UserName	Child element of UserNameToken element of SOAP Header	To validate against System Account
RequesterType	Child element of SOAP Body	To get a LDAP branch where System Account will be validated.
RequesterAuthenticationID	Child element of SOAP Body	To validate the DUNS number that is presented in this element.
Signer Certificate	Out-of-Band	To validate the signature
Intermediate SSL Certificate	Out-of-Band	SSL handshake
Self Signed SSL Certificate	Out-of-Band	SSL Handshake(Staging Only)

Table 4: SMT Validation Parameters

5.4.4 Validation Steps

Following are the authentication and validation sequence that happens on the SMT perimeter:

- SMT does 2-way SSL hand-shake with the requester's (Third-Party or REP) endpoint.
- SMT gets the requester's name or string identifying the requester by reading the HTTP Header variable.
- SMT performs Schema Validation.
- SMT validates the signature of the incoming request.
- SMT validates the system account of the requester using the HTTP header variable ENTITY_NAME, UserName from UserNameToken, and RequesterType elements.
- SMT validates the DUNS number of the requester using the HTTP header variable ENTITY_NAME, RequesterType and RequesterAuthenticationID elements

If all the steps are successful, the request is sent to SMT internal systems. In the event of failure at any step, the fault message is sent with an appropriate fault code.

6. SMT Web Service Interfaces

6.1 SMT Ad-hoc Meter Usage Interface

Figure 4 presents the High level system view diagram for SMT Ad-hoc Meter Usage Interface.

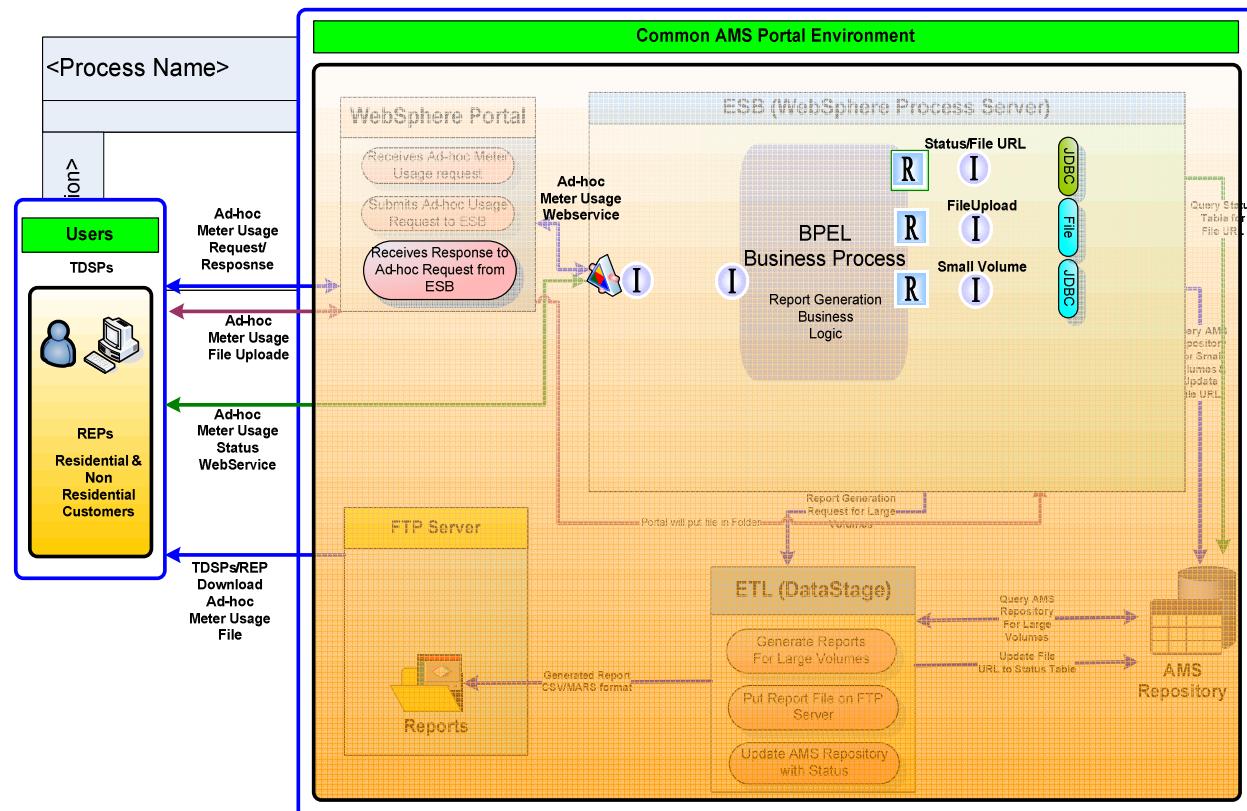


Figure 4: Ad-hoc Meter Usage Interface – High Level System View

This interface facilitates SMT Portal & Data Repository users to request & receive interval & register reads via machine to machine interface.

6.1.1 Interface Overview

- 1) An authorized user initiates an ad-hoc request in the web portal or invokes an ad-hoc Meter Usage Web service directly to retrieve usage info (interval read, register read or monthly billed usage data) for a given UserID, ESIIDs, supplied date, reportFormat (CSV or MARS) and reportType (Daily Register read or 15 minutes interval read or Both) parameters to see the latest or all the updates that has occurred to the data.

- 2) For the users logged in to portal UI, portal will build the request based on the user inputs, XSD and WSDL provided. Portal will invoke ad-hoc MeterUsage web service hosted on the ESB and pass request object in a SOAP envelope. For External users (TDSPs, REPs and Third-Parties) directly invoking the Web service, the SMT Datapower will do the security validations and then invoke ESB hosted Web service by passing the request object in a SOAP envelope.
- 3) ESB will map the incoming request object received from portal or external users to ASBO (Application Specific Business Object). Mapping is based on XSLT transformation. ASBO will contain all fields of request and common fields required for all AMS service that are going to take part in BPEL.
- 4) The BPEL deployed on the ESB contains the business logic to invoke appropriate service based on predefined criteria
- 5) For small volumes,
 - a) One input ESIID and up to 40 days of usage data: ESB fetches data from the SMT data repository and writes the output to a CSV file in shared folder. ESB then responds to the synchronous Web service request and returns the file url in SOAP response message to the portal or external user.
- 6) For Medium volumes,
 - a) One input ESIID and usage data greater than 40 days but less than 13 months and ReportType selected as CSV or MARS: ESB sets the priority based on input parameters and submits a message to a MQ queue to initiate an asynchronous request for the ETL components to fetch the data. The MQ message will have input parameters, message priority, correlationId generated by BPEL process and desired output file name.
- 7) For large volumes
 - a) More than one ESIID and ReportType selected as CSV or MARS: ESB sets the priority based on input parameters and submits a message to a MQ queue to initiate an asynchronous request for the ETL components to fetch the data. The MQ message will have input parameters, message priority; correlationId generated by BPEL process and desired output file name.
 - b) User uploaded ESIID/MeterId File and ReportType selected as CSV or MARS: Portal will call the ESB web service with the fileUrl ESB will read the file from the shared folder, parse the file content and validate the ESIID or MeterId in the file. ESB then sets the priority based on input parameters and submits a message to a MQ queue to initiate an asynchronous request for the ETL components to fetch the data. The MQ message will have input parameters, message priority; correlationId generated by BPEL process and desired output file name.

Table 5 shows the ad-hoc Meter Usage Message Volumes classification.

Message Volumes	Criteria	Invocation	Estimated Response Time
Small	1 ESIID and Usage data less than equal to 40 Days	Real-time Synchronous	Less than 3-5 minutes for 95% of the requests
Medium	1 ESIID and Usage date greater than 40 but less than equal to 13 Months	Batch & Asynchronous	2-3 hours for 95% of the requests
Large	More than 1 ESIID and Usage Data up to 4 years	Batch & Asynchronous	24 hours for 95% of the requests

Table 5: Ad-hoc Meter Usage Message Volumes

Note: The File upload functionality is not supported for External users calling ESB Web service directly.

8) Limitations on the number of ESIIDs per ad-hoc usage request

Table 6 shows the ad-hoc Meter Usage Message Limitations.

Number of ESIIDs	Number of days	Number of requests per day	Max Estimated Response Time All
1	365	20	24 hours
10	20	20	24 hours
50	4	20	24 hours
100	2	10	24 hours
200	1	10	24 hours

Table 6: Ad-hoc Meter Usage Message Limitations

§ *How to Read the Table*

- A Third-Party or REP would be able to raise one of the set of requests per day –
- 20 requests containing maximum 1 ESIIDs for the 365 days of data OR
- 20 requests containing maximum 10 ESIIDs for the 20 days of data OR
- 20 requests containing maximum 50 ESIIDs for the 5 days of data OR
- 10 requests containing maximum 100 ESIIDs for 2 days of data OR
- 10 requests containing maximum 200 ESIIDs for 1 day's data.

§ *Notes:*

- The intended use of ad-hoc API is for specific requests with a set of ESIIDs and for a defined period of time. The LSE usage files are sent daily to the REPs and the ad-hoc API is not advised to be used as a bulk backfill mechanism.
- The maximum number of ESIIDs per request = 200. This is a hard parameter
- Exceeding the parameters in the table (see table 3) may result in request being removed from queue. Notification will not be provided if this occurs.
- If a Third-Party or REP needs to make a large request for historical data that exceeds the above guidelines, it needs to submit a change request with AMIT at <http://www.puc.state.tx.us/industry/projects/electric/34610/34610.aspx>. The Operations Work Group can determine if and how to address the request.

9) For Medium and Large volumes scenarios # 6 (a), 7 (a) and 7 (b), ESB returns the SOAP response to the portal with the correlationId, statusCode and statusMessage. ETL processes the data and updates the Status table with userID, userType, correlationId, fileUrl, creationDate

10) ESB will host the Status web service to the give status response back to portal or External User.

- a) An authorized user initiates an ad-hoc status request in the web portal to retrieve status of MeterUsage request by calling ESB webservice(Status)
- b) The user request for status with UserID and correlationId(s).
- c) ESB will validate the UserID, correlationID association before fetching the status.

- d) If the validation is successful, ESB will fetch the data from the database for a given UserID & correlationId(s)
 - e) ESB will give response back to portal or external user with correlationId(s) and corresponding file name(s).
- 11) Compressed and encrypted output usage data file(s) and error report files (if any) are made available in the user specific directories for the requester to pull.
- a) Output data file(s) remains for 10 calendar days on FTPS site and will be deleted thereafter.

Table 7 shows the set of ad-hoc Meter usage interface requests, acknowledgements and responses.

Interface Message Description	Response	Invocation Type
Ad hoc Meter Usage Small Volume Request	Ad hoc Meter Usage Response	Synchronous
Ad hoc Meter Usage Medium Volume Request	Ad hoc Meter Usage Request Acknowledgement	Asynchronous
Ad hoc Meter Usage Large Volume Request	Ad hoc Meter Usage Request Acknowledgement	Asynchronous
Ad hoc Meter Usage Large Volume Request With file upload	Ad hoc Meter Usage Request Acknowledgement	Asynchronous
Ad hoc Meter Usage Status	Ad hoc Meter Usage Status Response	Synchronous

Table 7: Ad-hoc Meter Usage Request Invocation Pattern

6.1.2 Schema Definitions

All Ad-hoc Meter Usage requests use the following schema definition:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:met="http://schemas.esb.ams.com/meterusagesource">
```

6.1.3 Schema and Sample Messages

6.1.3.1 Ad-hoc Meter Usage Request

Table 8 shows the Request Schema elements.

Element	Mandatory	Type	Description
UserId	Y	string(100)	UserId of the requester
UserType	Y	string(10)	UserType of the requester (e.g REP,TDSP,RES,NON, and REG)
reportTypeArray	Y	Array	Report Type (Daily and/or Interval)
reportFormat	N	string(4)	Report Format (CSV or MARS)
ESIIDArray	Y	Array	Array of ESIID

Table 8: Ad-hoc Meter Usage Request Schema Elements

SOAP request schema example:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" 
  xmlns:met="http://schemas.esb.ams.com/meterusagesource">
  <soapenv:Header/>
  <soapenv:Body>
    <met:processMeterUsage>
      <MeterUsageReqList>
        <userId>JONBREZON</userId>
        <userType>REP</userType>
        <reportTypeArray>
          <!--Zero or more repetitions:-->
          <reportTypeArray>
            <reportType>INTERVAL</reportType>
            <startDate>07/01/2009</startDate>
            <endDate>08/09/2009</endDate>
          </reportTypeArray>
          <reportTypeArray>
            <reportType>DAILY</reportType>
            <startDate>07/01/2009</startDate>
            <endDate>08/09/2009</endDate>
          </reportTypeArray>
        </reportTypeArray>
        <!--Optional :-->
        <reportFormat>CSV</reportFormat>
        <ESIIDArray>
          <!--1 or more repetitions:-->
          <ESIID>1008901022900060000000</ESIID>
        </ESIIDArray>
      </MeterUsageReqList>
    </met:processMeterUsage>
  </soapenv:Body>
</soapenv:Envelope>
```

6.1.3.2 Ad-hoc Meter Usage Request with File Upload

Table 9 shows the Request Schema elements with File Upload.

Element	Mandatory	Type	Description
UserId	Y	string	UserId of the requester
UserType	Y	string	UserType of the requester (e.g. REP,TDSP,RES,NON, and REG)

Element	Mandatory	Type	Description
reportTypeArray	Y	Array	Report Type (Daily and/or Interval)
reportFormat	N	string	Report Format (CSV or MARS)
FileUrl	Y	String	Url of user uploaded file

Table 9: Ad-hoc Meter Usage with Uploaded File Request Schema Elements

SOAP request schema example:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:met="http://schemas.esb.ams.com/meterusagesource">
  <soapenv:Header/>
  <soapenv:Body>
    <met:processMeterUsageFile>
      <MeterUsageFileReq>
        <userId>JONBREZON</userId>
        <userType>RES</userType>
        <reportTypeArray>
          <!--Zero or more repetitions:-->
          <reportTypeArray>
            <reportType>INTERVAL</reportType>
            <startDate>06/29/2009</startDate>
            <endDate>06/29/2009</endDate>
          </reportTypeArray>
        </reportTypeArray>
        <!--Optional:-->
        <reportFormat>CSV</reportFormat>
        <fileUrl> /data/AdhocMeterUsage/ESB/INPUT/DailyMeterUsage6e05214d0b4d5e700c96fd0f.CSV
      </fileUrl>
      </MeterUsageFileReq>
    </met:processMeterUsageFile>
  </soapenv:Body>
</soapenv:Envelope>
```

6.1.3.3 Ad-hoc Meter Usage Response

Table 10 shows the Response Schema elements.

Element	Mandatory	Type	Description
CorrelationId	Y	string(100)	UserId of the requester
statusCode	N	string(10)	UserType of the requester
statusMessage	N	Array	Report Type (Daily and/or Interval)

Table 10: Ad-hoc Meter Response Schema Elements

SOAP response schema example:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" 
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" 
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Header/>
    <soapenv:Body>
        <me:processMeterUsageResponse xmlns:me="http://schemas.esb.ams.com/meterusagesource">
            <MeterUsageResponse xsi:nil="true"/>
            <MeterUsageSOAPResponse>
                <correlationId>f84da12ccbafc7dd16603909</correlationId>
                <statusCode>0</statusCode>
                <statusMessage>Data record completed successfully</statusMessage>
            </MeterUsageSOAPResponse>
        </me:processMeterUsageResponse>
    </soapenv:Body>
</soapenv:Envelope>

```

6.1.3.4 Ad-hoc Meter Usage Report File Naming Convention

Table 11 shows the Ad hoc Meter Usage File Naming Convention.

Element	Explanation	Format
ReportName	IntervalMeterUsage/ DailyMeterUsage	Alpha(20)
CorrelationID	CorrelationID for specified request	Alphanumeric (32)
.lse / .csv	Extension depending on file format CSV or MARS (lse)	Alpha (3)
DUNSNumber	Optional DUNS Number	Numeric(13)

Table 11: Ad-hoc Meter Usage Report File Request Naming Convention

IntervalMeterUsage<CorrelationID>.lse/.csv<OptionalDUNSNumber>

DailyMeterUsage<CorrelationID>.csv<OptionalDUNSNumber>

6.1.3.5 Ad-hoc Meter Usage Report File Format

Interval Meter Usage Report:

Suggested format:

Comma delimited file, with first line being field names.

Fields:

- ESI ID (Varchar 64)
- Time Stamp Start (date time stamp)
- Time Stamp End (date time stamp)
- Metered KWH (decimal)

File name format:

File Name will be generated by ESB with following pattern.

IntervalMeterUsage<CorrelationID>.lse/.csv.<OptionalDUNSNumber>

Ex: IntervalMeterUsage f84da12ccbafc7dd16603909.csv. 957877905

Ex:

File Format:

ESI ID, Time Stamp Start, Time Start End, Metered KWH, Status

Ex:

12345678909876543, 2009-05-22T12:00:00, 2009-05-22T12:15:00, 1.5

Daily Meter Usage Report:

Suggested format:

Comma delimited file, with first line being field names.

Fields:

- ESI ID (Varchar 64)
- Time Stamp (date time stamp)
- Metered KWH (decimal)

File name format:

File Name will be generated by ESB with following pattern.

DailyMeterUsage<CorrelationID>.csv.<OptionalDUNSNumber>

Ex: DailyMeterUsage f84da12ccbafc7dd16603909.csv. 957877905

File Format:

ESI ID, Time Stamp Start, Time Start End, Metered KWH, Status

Ex:

12345678909876543, 2009-05-22, 2000,

[6.1.3.6 Ad-hoc Meter Usage Report Status Request](#)

Table 12 shows the Status Request Schema elements.

Element	Explanation	Format
UserId	UserId of the requester	String
CorrelationID	CorrelationID for specified request	String

Table 12: Ad-hoc Meter Usage Report Status Request Schema Elements

SOAP request schema example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
    xmlns:met="http://schemas.esb.ams.com/meterusagesource">  
    <soap:Header/>  
    <soap:Body>  
        <met:meterUsageStatus>  
            <statusRequest>  
                <userId> JONBREZON </userId>  
                <correlationId> f84da12ccbafc7dd16603909</correlationId>  
            </statusRequest>  
        </met:meterUsageStatus>  
    </soap:Body>  
</soap:Envelope>
```

6.1.3.7 Ad-hoc Meter Usage Report Status Response

Table 13 shows the Status response Schema elements.

Element	Explanation	Format
correlationId	ESB generated correlationId	String
fileUrl	File url of the generated report	String
status	Report generation status	String

Table 13: Ad-hoc Meter Usage Report Status Response Schema Elements

SOAP request schema example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
    xmlns:met="http://schemas.esb.ams.com/meterusagesource">  
    <soap:Header/>  
    <soap:Body>  
        <met:meterUsageStatus>  
            <BO_MeterUsageStatusResList>  
                <UsageStatusResponse>  
                    <correlationId> f84da12ccbafc7dd16603909</correlationId>  
                    <fileUrl />  
                    <status>sucess</status>
```

```
</UsageStatusResponse>
</BO_MeterUsageStatusResList>
</met:meterUsageStatus>
</soap:Body>
</soap:Envelope>
```

6.2 SMT Re-branding Interface

The Re-branding API is envisioned to be used by Third-Parties and REPs Portal Systems to aggregate the graphical user interface with specific branding requirement. The API only provides data required to build the portal around it. The client systems are entirely responsible for rendering branding information/content on graphical interface of system.

This API enables:

- Daily Usage report
- Interval Usage report
- Monthly Billed Usage report

6.2.1 Interface Overview

The web service requests are intercepted by SMT Data Power (SMT-DP) component, which handles security for requests coming into SMT. SMT-DP authorizes requests to be forwarded to back-end Enterprise Service Bus of SMT (SMT-ESB). SMT-DP is also responsible for validating schema compliance of incoming requests. In any of un-satisfied requirement of requests, SMT-DP rejects the request through SOAP Faults.

SMT-ESB processes the requests as the business logic and generates the required usage reports and sends as response to its clients. Any failure on SMT-ESB is reported via SOAP Faults. Currently SMT-ESB supports three types of reports as part of the Re-branding web service.

1. DAILY : This report returns daily usage of energy consumption in KWH for a Meter or ESIID
2. INTERVAL : This report returns interval usage of energy consumption in KWH between the dates specified the request for a Meter or ESIID
3. MONTHLY : This report returns monthly usage of energy consumption in KWH between the dates specified the request for a Meter or ESIID

The Re-branding web service will respond with one of the following messages:

DailySyncResponse - When ReportType = DAILY

IntervalSyncResponse - When ReportType = INTERVAL

MonthlySyncResponse - When ReportType = MONTHLY

SMT generates reports for Daily & Interval types for one ESIID and up to 40 days of usage data. For Monthly usage report, SMT accepts one ESIID and up to 24 months of usage data. If any of the above condition is not satisfied, SMT generates SOAP Fault.

Rebranding

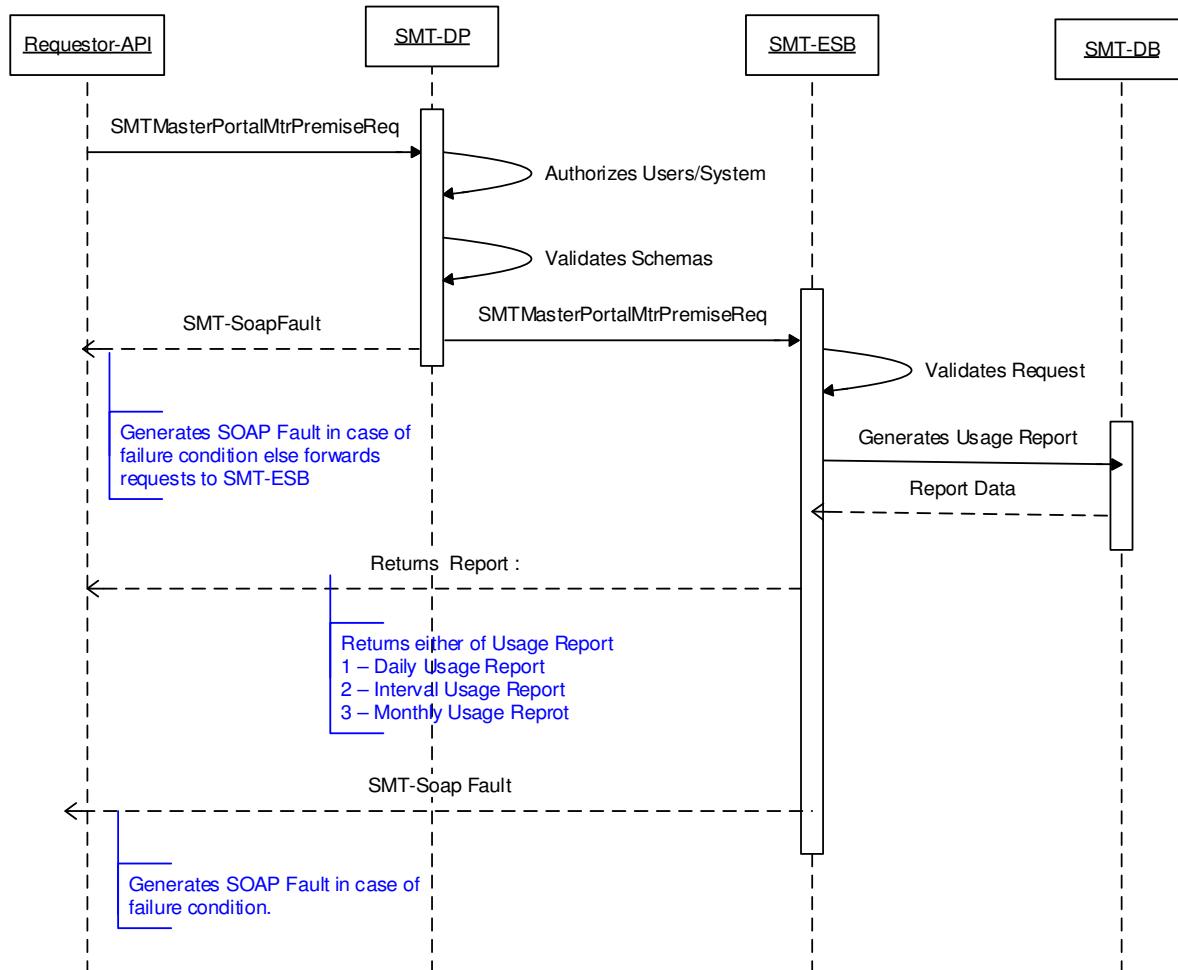


Figure 5: Re-branding Overall Interaction Diagram

6.2.2 Schema Definitions

All Re-branding requests use the following schema definition:

```
<soapenv:Envelope xmlns:met="http://schemas.esb.ams.com/meterusagesource"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
```

6.2.3 Schema and Sample Messages

6.2.3.1 SMT Re-branding Interface Definition Information

SMT Re-branding interface (WSDL) has three service operations viz.

- *processMeterUsage:*

The service is used design to generate synchronous responses for small set of volumes and asynchronous response (file based response) to large volume. The service supports Daily, Interval & Monthly type usage reports.

This service is not described in this document, though this part of WSDL definition. The service is currently consumed by SMT internal components. Technically its possible that the Third-Parties and REPs use this for integration.

- *processMeterUsageFile:*

The service supports file based request, where-in clients can upload the file with list of ESIID's / Meters for generating the report.

This service is not described in this document, though this part of WSDL definition. The service is currently consumed by SMT internal components. Currently file based operations are not supported through SMT API.

- *processREPRBranding:*

The service dedicated for the re-branding requirement and is exposed to Thrid-Parties and REPs for integrating with SMT system for branding of their local portal graphical user interface. The service interaction & message are described in detail in rest of the sections of this document.

The screenshot shows a software interface with a toolbar at the top and a main panel titled 'Operations'. Below the title, it says 'Operations and their parameters'. A tree view on the left lists three operations: 'processMeterUsage', 'processMeterUsageFile', and 'processREPRBranding'. Each operation has an expandable section showing 'Input(s)', 'Output(s)', and 'Fault' details. A table below the tree view summarizes these details.

Operations		
Operations and their parameters		
	Name	Type
processMeterUsage		
Input(s)	MeterUsageReqList	BO_MeterUsageRequest
Output(s)	MeterUsageResponse	BO_MeterUsageFileResponse
Fault	MeterUsageFaultResponse	SOAPFault
processMeterUsageFile		
Input(s)	MeterUsageFileReq	BO_MeterUsageFileRequest
Output(s)	MeterUsageSOAPResponse	BO_SOAPResponse
Fault	MeterUsageFaultResponse	SOAPFault
processREPRBranding		
Input(s)	REPRBrandingRequest	BO_REPRBrandingReq
	IntervalSyncResponse	BO_MeterUsageInFileWrite
Output(s)	DailySyncResponse	BO_MeterUsageDailFileWrite
	MonthlySyncResponse	BO_MeterUsageMnlyFileWrite
Fault	FaultResponse	SOAPFault

Figure 6: Re-branding Interaction Operations

6.2.3.2 SMT Re-branding Request

The SMT Re-branding API supports DAILY and/or INTERVAL and/or MONTLY billed usage reports.

When sending a device provisioning request, the SMT request will include:

The Security information in SOAP header as described in section 5 of this document.

Re-branding request information formatted as described in Table 14.

Element	Mandatory	Type	Description
userId	N	string(100)	SMT System Account ID of the requester.
userType	Y	string(10)	Requester Type indicator. Accepted values are: REP - Retail Electric Provider; TDSP - Transmission & Distribution Service Provider; RES - Residential Customer; NRC - Non residential Customer;
reportFormat	N	string	A response format of report. Currently XML report is supported. Accepted values are: XML xml
ReportType	Y	string	Type of report/response. Accepted values are: DAILY - daily usage report from the date INTERVAL - interval usage report from startDate to endDate. MONTHLY - monthly usage report from startDate to endDate
startDate	Y	string(10)	Start date of the report
endDate	Y	string(10)	End date of the report
ESIID	Y	string(17,64)	Energy Service Interface identifier

Table 14: Re-branding Request Schema Elements

A sample re-branding request appears below.

```

<soapenv:Envelope xmlns:met="http://schemas.esb.ams.com/meterusagesource"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body wsu:Id="id-13060949" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
  200401-wss-wssecurity-utility-1.0.xsd">
    <met:processREPReBranding>
      <REPReBrandingRequest>
        <userId>TP_USR_ID</userId>
        <userType>THRD</userType>
        <reportTypeArray>
          <!--1 or more repetitions:-->
          <reportTypeArray>
            <reportType>DAILY</reportType>
            <startDate>01/01/2013</startDate>
            <endDate>01/31/2013</endDate>
          </reportTypeArray>
        </reportTypeArray>
      </REPReBrandingRequest>
    </met:processREPReBranding>
  </soapenv:Body>
</soapenv:Envelope>

```

```

<!--Optional:-->
<ESIIDArray>
  <!--1 or more repetitions:-->
  <ESIID>10443720003671954</ESIID>
</ESIIDArray>
</REPReBrandingRequest>
</met:processREPReBranding>
</soapenv:Envelope>

```

6.2.3.3 SMT Re-branding Daily Sync Response

Table 15 shows the Re-branding Daily Sync Response Schema elements.

Element	Mandatory	Type	Description
ESIID	Y	string(17,64)	Energy Service Interface identifier
usageDate	N	string	Usage date
meteredKWH	N	string(30)	Metered KWH value
START_READING	N	string	Reading start date
END_READING	N	string	Reading end date

Table 15: SMT Re-branding Daily Sync Response Schema Elements

Following XML is a sample of the SOAP response for Daily Sync Response message:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header/>
  <soapenv:Body>
    <me:processREPReBrandingResponse
      xmlns:me="http://schemas.esb.ams.com/meterusagesource">
      <IntervalSyncResponse/>
      <DailySyncResponse>
        <MeterUsageDailFile>
          <ESIID>10443720003671954</ESIID>
          <usageDate>01/29/2013</usageDate>
          <meteredKWH>37.382</meteredKWH>
          <START_READING>23116.464</START_READING>
          <END_READING>23153.84</END_READING>
        </MeterUsageDailFile>
        <MeterUsageDailFile>
          <ESIID>10443720003671954</ESIID>
          <usageDate>01/30/2013</usageDate>

```

```

<meteredKWH>57.529</meteredKWH>
<START_READING>23153.84</START_READING>
<END_READING>23211.368</END_READING>
</MeterUsageDailFile>
<MeterUsageDailFile>
<ESIID>10443720003671954</ESIID>
<usageDate>01/31/2013</usageDate>
<meteredKWH>63.377</meteredKWH>
<START_READING>23211.368</START_READING>
<END_READING>23274.746</END_READING>
</MeterUsageDailFile>
</DailySyncResponse>
<MonthlySyncResponse/>
</me:processREPRReBrandingResponse>
</soapenv:Body>
</soapenv:Envelope>

```

6.2.3.4 SMT Re-branding Interval Sync Response

Table 16 shows the Re-branding Daily Sync Response Schema elements.

Element	Mandatory	Type	Description
ESIID	Y	string(17,64)	Energy Service Interface identifier
timeStampStart	N	string	Start Date for request
timeStampEnd	N	string(30)	End Date for request
meteredKWH	N	string	Metered KWH value
Act_Est	N	string	

Table 16: SMT Re-branding Interval Sync Response Schema Elements

Following XML is a sample of the SOAP response for Interval Sync Response message:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <ns1:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:xsd="http://www.w3.org/2001/XMLSchema">
  <ns1:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Header/>
    <soapenv:Body>
      <me:processREPRReBrandingResponse
        xmlns:me="http://schemas.esb.ams.com/meterusagesource">
        <IntervalSyncResponse>
          <MeterUsageIntFile>
            <ESIID>1008901002100766042100</ESIID>
            <timeStampStart>01/01/2010 12:00 am</timeStampStart>

```

```
<timeStampEnd>01/01/2010 12:15 am</timeStampEnd>
<meteredKWH>31.47</meteredKWH>
<Act_Est>E</Act_Est>
</MeterUsageIntFile>
<MeterUsageIntFile>
<ESIID>1008901002100766042100</ESIID>
<timeStampStart>01/01/2010 12:15 am</timeStampStart>
<timeStampEnd>01/01/2010 12:30 am</timeStampEnd>
<meteredKWH>17.68</meteredKWH>
<Act_Est>A</Act_Est>
</MeterUsageIntFile>
<MeterUsageIntFile>
<ESIID>1008901002100766042100</ESIID>
<timeStampStart>01/01/2010 12:30 am</timeStampStart>
<timeStampEnd>01/01/2010 12:45 am</timeStampEnd>
<meteredKWH>35.25</meteredKWH>
<Act_Est>A</Act_Est>
</MeterUsageIntFile>
<MeterUsageIntFile>
<ESIID>1008901002100766042100</ESIID>
<timeStampStart>01/01/2010 12:45 am</timeStampStart>
<timeStampEnd>01/01/2010 01:00 am</timeStampEnd>
<meteredKWH>.82</meteredKWH>
<Act_Est>A</Act_Est>
</MeterUsageIntFile>
<MeterUsageIntFile>
<ESIID>1008901002100766042100</ESIID>
<timeStampStart>01/01/2010 01:00 am</timeStampStart>
<timeStampEnd>01/01/2010 01:15 am</timeStampEnd>
<meteredKWH>57.8</meteredKWH>
<Act_Est>A</Act_Est>
</MeterUsageIntFile>
<MeterUsageIntFile>
<ESIID>1008901002100766042100</ESIID>
<timeStampStart>01/01/2010 01:15 am</timeStampStart>
<timeStampEnd>01/01/2010 01:30 am</timeStampEnd>
<meteredKWH>83.58</meteredKWH>
<Act_Est>E</Act_Est>
</MeterUsageIntFile>
</IntervalSyncResponse>
<DailySyncResponse/>
<MonthlySyncResponse/>
```

```

</me:processREPRReBrandingResponse>
</soapenv:Body>
</soapenv:Envelope>
```

6.2.3.5 SMT Re-branding Monthly Sync Response

Table 17 shows the Re-branding Daily Sync Response Schema elements.

Element	Mandatory	Type	Description
ESIID	Y	string(17,64)	Energy Service Interface identifier
ACTL_STRT_DT	N	string	Actual Start Date
ACTL_END_DT	N	string	Actual End Date
ACTL_KWH_USG	N	string	Actual KWH Usage
MTRD_KWH_USG	N	string	Metered KWH Usage
BLLD_KWH_USG	N	string	Billed KWH Usage
MTRD_KVA	N	string	Metered KVA
BLLD_KVA	N	string	Billed KVA

Table 17: SMT Re-branding Monthly Sync Response Schema Elements

Following XML is a sample of the SOAP response for Monthly Sync Response message:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header/>
  <soapenv:Body>
    <me:processREPRReBrandingResponse
      xmlns:me="http://schemas.esb.ams.com/meterusagesource">
      <IntervalSyncResponse/>
      <DailySyncResponse>
        <MeterUsageDailFile>
          <ESIID>1008901002100766042100</ESIID>
          <usageDate>01/01/2010</usageDate>
          <meteredKWH>10030.93</meteredKWH>
          <START_READING>9942</START_READING>
          <END_READING>9944</END_READING>
        </MeterUsageDailFile>
        <MeterUsageDailFile>
          <ESIID>1008901002100766042100</ESIID>
          <usageDate>01/02/2010</usageDate>
          <meteredKWH>10030.93</meteredKWH>
          <START_READING>9942</START_READING>
```

```
<END_READING>9944</END_READING>
</MeterUsageDailFile>
<MeterUsageDailFile>
<ESIID>1008901002100766042100</ESIID>
<usageDate>01/03/2010</usageDate>
<meteredKWH>10030.93</meteredKWH>
<START_READING>9942</START_READING>
<END_READING>9944</END_READING>
</MeterUsageDailFile>
</DailySyncResponse>
<MonthlySyncResponse>
<MeterUsageMnthalFile>
<ESIID>1008901002100766042100</ESIID>
<ACTL_STRT_DT>02/03/2006</ACTL_STRT_DT>
<ACTL_END_DT>03/01/2006</ACTL_END_DT>
<ACTL_KWH_USG>4000</ACTL_KWH_USG>
<MTRD_KWH_USG>0</MTRD_KWH_USG>
<BLDD_KWH_USG>0</BLDD_KWH_USG>
<MTRD_KVA>0</MTRD_KVA>
<BLDD_KVA>0</BLDD_KVA>
</MeterUsageMnthalFile>
<MeterUsageMnthalFile>
<ESIID>1008901002100766042100</ESIID>
<ACTL_STRT_DT>01/03/2006</ACTL_STRT_DT>
<ACTL_END_DT>02/01/2006</ACTL_END_DT>
<ACTL_KWH_USG>4000</ACTL_KWH_USG>
<MTRD_KWH_USG>0</MTRD_KWH_USG>
<BLDD_KWH_USG>0</BLDD_KWH_USG>
<MTRD_KVA>0</MTRD_KVA>
<BLDD_KVA>0</BLDD_KVA>
</MeterUsageMnthalFile>
</MonthlySyncResponse>
</me:processREPRReBrandingResponse>
</soapenv:Body>
</soapenv:Envelope>
```

6.3 SMT Third-Party Interfaces

The SMT Third-Party interfaces provide the Third-Parties the ability to:

- Initiate a new Energy Data Agreement
- Request an extension of an active Energy Data Agreement
- Initiate a new In-Home Device Services Agreement
- Initiate a new In-Home Device Agreement
- Request to re-add an In-Home Device with an active In-Home Device Agreement
- Request Scheduled Usage Reports

6.3.1 Interface Overview

When SMT receives an On Demand Read Request, it performs the following validations:

- XML validation is performed by an appliance on the SMT perimeter. If the request fails validation, a SOAP fault is returned.
- If XML is formatted correctly, the following business validations will be performed by SMT:
 - The Account Authorization Code provided is valid
 - The email address provided is not being used by a registered SMT customer if the agreement is intended for an unregistered customer
 - The ESIID(s) provided is valid
 - The ESIID has not reached the maximum of 5 In-Home devices (for In-Home Device Agreement only)
 - The In-Home Device(s) provided is not being provisioned to another ESIID (for In-Home Device Agreement only)
 - There is no other active In-Home Device Services Agreements for the same In-Home Device(s) provided

If the request fails any of these validations, an error return code and return message will be provided in the Request Acknowledgement indicating that the request is not accepted for processing.

Once the request is validated, the status of the request will be set to Acknowledged (ACK) and the response acknowledgement will be sent to the requester.

6.3.2 Schema Definitions

The Third-Party APIs requests use the following schema definitions:

- Energy Data Agreement Request

```
<soapenv:Envelope xmlns:bim="http://schemas.esb.ams.com/BIM_SMTxPEnergyDataAgreement"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
```

- Energy Data Agreement Extension Request

```
<soapenv:Envelope xmlns:bim="http://schemas.esb.ams.com/BIM_SMTxPEnergyDataAgreement"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
```

- In-Home Device Services Agreement Request

```
<soapenv:Envelope xmlns:bim="http://schemas.esb.ams.com/BIM_SMTxPHANServicesAgreement"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
```

- In-Home Device Agreement Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:inf="http://BIM_SMTxPHANDeviceAgreement_Library/INF_SMTxPHANDeviceAgreement">
```

- Re-add In-Home Device Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:inf="http://BIM_SMTxPHANDeviceAgreement_Library/INF_SMTxPHANDeviceAgreement">
```

- Scheduled Usage Report Request

```
<soapenv:Envelope xmlns:bim="http://schemas.esb.ams.com/BIM_BIM_SMTxPScheduledReport"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
```

6.3.3 Schema and Sample Messages

6.3.3.1 Energy Data Agreement Request

Table 18 shows the Third-Party Energy Data Agreement Request Schema elements.

Element	Mandatory	Type	Description
SMTRegistered	Y	Int	If SMT = 1 (Unregistered) SMT email address is required; If SMT = 2 (Registered), Account Authorization Code is required
CustomerType	Y	int	Energy Service Interface identifier
AccountAuthorizationCode	N	string(8)	Customer Type: 1 (Residential) or 2 (Business)
SMTEmail	N	string(25)	Provides authorized 3rd Parties the capability to initiate Energy Data Agreement
FirstName	N	string(25)	SMT Email Address
LastName	N	string(25)	If SMT Email Address is provided, SMT will use First Name from customer's profile
PhoneNumber	N	string(10)	If SMT Email Address is provided, SMT will use Last Name from customer's profile
LanguagePreference	N	int	If SMT Email Address is provided, SMT will use Phone Number from customer's profile

Element	Mandatory	Type	Description
CompanyName	Y	string(25)	SMT is the system of record and we use the value available at SMT, when 3rd P inputs a different value.
ContactName	N	string(25)	This should be optional. SMT will use the Contact Name from 3rd Party's company profile if no value was entered for this field
ContactPhone	N	string(10)	This should be optional. SMT will use the Contact Name from 3rd Party's company profile if no value was entered for this field
ContactEmail	N	string(25)	This should be optional. SMT will use the Contact Name from 3rd Party's company profile if no value was entered for this field
AgreementDuration	Y	int	Agreement Duration Options: 1 = 1 month 2 = 3 months 3 = 6 months 4 = 9 months 5 = 12 months These are valid values for API but the back end business validation will ensure the appropriate value is selected.
FlexOrFixed	Y	int	1 = Flex; 2 = Fixed; For Flex we need one Esiid and for Fixed it should be one, some or all ESIIDS.
ESIID	Y	String (17,64)	Energy Service Interface identifier; If FlexOrFixed = 1, 1 ESI ID is required;
Comments	N	string(25)	This is optional.
Terms	Y	string(3)	Terms and Conditions = only accepts value "Yes" - (case insensitive)

Table 18: Third-Party Energy Data Agreement Request Schema Elements

Following XML is a sample of the SOAP message of the Energy Data Agreement Request:

```

<soapenv:Envelope xmlns:bim="http://schemas.esb.ams.com/BIM_SMTxPEnergyDataAgreement"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body wsu:Id="id-33158755" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
  200401-wss-wssecurity-utility-1.0.xsd">
    <bim:processEnergyDataAgreement>
      <SMTxPEnergyDataAgreementRequest>
        <!--Optional:-->
        <RequestID>1234</RequestID>
        <RequesterType>3</RequesterType>
        <RequesterAuthenticationID>860700001</RequesterAuthenticationID>
        <RequesterID>JF_3RDPARTY1_ADMIN1</RequesterID>
        <RequestPriority>M</RequestPriority>
        <!--Optional:-->
      </SMTxPEnergyDataAgreementRequest>
    </bim:processEnergyDataAgreement>
  </soapenv:Body>
</soapenv:Envelope>

```

```
<CallbackUri></CallbackUri>
<CustomerAgreementConfigMessage>
  <MessageHeader>
    <!--Optional:-->
    <verb>Create</verb>
    <!--Optional:-->
    <noun>CustomerAgreementConfig</noun>
  </MessageHeader>
  <MessagePayload>
    <CustomerAgreementConfig>
      <SMTRegistered>2</SMTRegistered>
      <CustomerType>1</CustomerType>
      <!--Optional:-->
      <AccountAuthorizationCode>1Q2W3E4R</AccountAuthorizationCode>
      <!--Optional:-->
      <SMTEmail>CustomerEmail@yahoo.com</SMTEmail>
      <!--Optional:-->
      <FirstName>John</FirstName>
      <!--Optional:-->
      <LastName>Hunt</LastName>
      <!--Optional:-->
      <PhoneNumber>9099099099</PhoneNumber>
      <!--Optional:-->
      <LanguagePreference>2</LanguagePreference>
      <ServiceSupplier>
        <CompanyName>Third Party Adm 055</CompanyName>
        <!--Optional:-->
        <ContactName>NonResSpanishUser2</ContactName>
        <!--Optional:-->
        <ContactPhone>2142142144</ContactPhone>
        <!--Optional:-->
        <ContactEmail/>
      </ServiceSupplier>
      <ValidityInterval>
        <AgreementDuration>2</AgreementDuration>
        <FlexOrFixed>1</FlexOrFixed>
      </ValidityInterval>
      <ServiceDeliveryPoint>
        <!--1 to 200 repetitions:-->
        <ESIID>1008901012126335709100</ESIID>
      </ServiceDeliveryPoint>
      <!--Optional:-->
      <Comments>Your comments goes here..</Comments>
      <Terms>Yes</Terms>
```

```

</CustomerAgreementConfig>
</MessagePayload>
</CustomerAgreementConfigMessage>
</SMTxPEnergyDataAgreementRequest>
</bim:processEnergyDataAgreement>
</soapenv:Body>
</soapenv:Envelope>

```

6.3.3.2 Energy Data Agreement Request Acknowledgement

Table 19 shows the Third-Party Energy Data Agreement Request Acknowledgment Schema elements.

Element	Mandatory	Type	Description
RequestID	Y	string(32)	Unique request identifier. This will be generated by SMTxP & sent back in acknowledgement by TDSP
RequestStatus	N	string(3)	Status of the request. Values: ACK - Acknowledgement indicating the request was accepted for further processing. FLR - Failure, indicating the request was not accepted for processing
RequestStatusDesc	N	String(64)	Description of the request status code.
ESIID	Y	string(17,64)	Energy Service Interface Identifier
Reason	N	string(128)	Reason of Failure at individual ESIID

Table 19: Third-Party Energy Data Agreement Request Acknowledgement Schema Elements

Following XML is a sample of the SOAP message of the Energy Data Agreement Request Acknowledgement:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header/>
  <soapenv:Body>
    <ag:processEnergyDataAgreementResponse
      xmlns:ag="http://schemas.esb.ams.com/BIM_SMTxPEnergyDataAgreement">
      <SMTxPEnergyDataAgreementAck>
        <RequestID>1234</RequestID>
        <AgreementNumber>3487</AgreementNumber>
        <RequestStatus>ACK</RequestStatus>
        <RequestStatusDesc>Success</RequestStatusDesc>
      </SMTxPEnergyDataAgreementAck>
    </ag:processEnergyDataAgreementResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

6.3.3.3 Energy Data Agreement Extension Request

Table 20 shows the Third-Party Energy Data Agreement Request Schema elements.

Element	Mandatory	Type	Description
AgreementNumber	Y	string(12)	The Energy Data Agreement Number.
AgreementDuration	Y	int	<p>Agreement Duration Options: Values 1 = 1 month 2 = 3 months 3 = 6 months 4 = 9 months 5 = Up to 12 months</p> <p>These are valid values for API but the back end business validation will ensure the appropriate value is selected. Example: If the Agreement has a duration of 10 months, the following options will be available in the dropdown: 1 month Up to 12 months</p>

Table 20: Third-Party Energy Data Agreement Extension Request Schema Elements

Following XML is a sample of the SOAP message of the Energy Data Agreement Extension Request:

```

<soapenv:Envelope xmlns:bim="http://schemas.esb.ams.com/BIM_SMTxPEnergyDataAgreement"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body wsu:Id="id-24336074" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
  200401-wss-wssecurity-utility-1.0.xsd">
    <bim:processEnergyDataAgreementExtension>
      <SMTxPEnergyDataAgreementExtRequest>
        <!--Optional:-->
        <RequestID>1234</RequestID>
        <RequesterType>3</RequesterType>
        <RequesterAuthenticationID>860700001</RequesterAuthenticationID>
        <RequesterID>JF_3RDPARTY1_ADMIN1</RequesterID>
        <RequestPriority>M</RequestPriority>
        <!--Optional:-->
        <CallbackUri/>
        <!--Optional:-->
        <CustomerAgreementConfigMessage>
          <MessageHeader>
            <verb>Create</verb>
            <noun>CustomerAgreementConfig</noun>
          </MessageHeader>
          <MessagePayload>
            <CustomerAgreementConfig>
              <CustomerAgreementExtension>
```

```

<AgreementNumber>3487</AgreementNumber>
<AgreementDuration>2</AgreementDuration>
</CustomerAgreementExtension>
</CustomerAgreementConfig>
</MessagePayload>
</CustomerAgreementConfigMessage>
</SMTxPEnergyDataAgreementExtRequest>
</bim:processEnergyDataAgreementExtension>
</soapenv:Body>
</soapenv:Envelope>

```

6.3.3.4 Energy Data Agreement Extension Request Acknowledgement

Table 21 shows the Third-Party Energy Data Agreement Extension Request Acknowledgment Schema elements.

Element	Mandatory	Type	Description
RequestID	Y	string(32)	Unique request identifier. This will be generated by SMTxP & sent back in acknowledgement by TDSP
RequestStatus	N	string(3)	Status of the request. Values: ACK - Acknowledgement indicating the request was accepted for further processing. FLR - Failure, indicating the request was not accepted for processing
RequestStatusDesc	N	String(64)	Description of the request status code.
AgreementNumber	Y	string(12)	Energy Data Agreement Number
Reason	N	string(128)	Reason of Failure at individual ESIID

Table 21: Third-Party Energy Data Agreement Extension Request Acknowledgement Schema Elements

Following XML is a sample of the SOAP message of the Energy Data Agreement Extension Request Acknowledgement:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ag:processEnergyDataAgreementExtensionResponse
      xmlns:ag="http://schemas.esb.ams.com/BIM_SMTxPEnergyDataAgreement">
      <SMTxPEnergyDataAgreementExtAck>
        <RequestID>1234</RequestID>
        <RequestStatus>ACK</RequestStatus>
        <RequestStatusDesc>Success</RequestStatusDesc>
      </SMTxPEnergyDataAgreementExtAck>
    </ag:processEnergyDataAgreementExtensionResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

```

</ag:processEnergyDataAgreementExtensionResponse>
</soapenv:Body>
</soapenv:Envelope>

```

6.3.3.5 In-Home Device Services Agreement Request

Table 22 shows the Third-Party In-Home Device Services Agreement Request Schema elements.

Element	Mandatory	Type	Description
SMTRegistered	Y	Int	If SMT = 1 (Unregistered) SMT email address is required; If SMT = 2 (Registered), Account Authorization Code is required
CustomerType	Y	int	Energy Service Interface identifier
AccountAuthorizationCode	N	string(8)	Customer Type: 1 (Residential) or 2 (Business)
SMTEmail	N	string(25)	Provides authorized 3rd Parties the capability to initiate Energy Data Agreement
FirstName	N	string(25)	SMT Email Address
LastName	N	string(25)	If SMT Email Address is provided, SMT will use First Name from customer's profile
PhoneNumber	N	string(10)	If SMT Email Address is provided, SMT will use Last Name from customer's profile
LanguagePreference	N	int	If SMT Email Address is provided, SMT will use Phone Number from customer's profile
CompanyName	Y	string(25)	SMT is the system of record and we use the value available at SMT, when 3rd P inputs a different value.
ContactName	N	string(25)	This should be optional. SMT will use the Contact Name from 3rd Party's company profile if no value was entered for this field
ContactPhone	N	string(10)	This should be optional. SMT will use the Contact Name from 3rd Party's company profile if no value was entered for this field
ContactEmail	N	string(25)	This should be optional. SMT will use the Contact Name from 3rd Party's company profile if no value was entered for this field
ESIID	Y	String (17,64)	Energy Service Interface identifier
DeviceMACAddr	Y	string(16)	Device MAC Address
MessageType	Y	String(5)	1=Simple text; 2=Load Control; 3=Pricing Signal If customer wants to choose Simple text, Load control and Price they will have to specify it as "1 2 3". For only Simple text & Load control it is "1 2", for Load control and Price its "2 3" and Simple text and Price its "1 3".
Comments	N	string(25)	This is optional.

Element	Mandatory	Type	Description
Terms	Y	string(3)	Terms and Conditions = only accepts value "Yes" - (case insensitive)

Table 22: Third-Party In-Home Device Services Agreement Request Schema Elements

Following XML is a sample of the SOAP message of the In-Home Device Services Agreement Request:

```

<soapenv:Envelope xmlns:bim="http://schemas.esb.ams.com/BIM_SMTxPHANServicesAgreement"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body wsu:Id="id-32518610" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd">
    <bim:processHANServicesAgreement>
      <SMTxPHANServicesAgreementRequest>
        <!--Optional:-->
        <RequestID/>
        <RequesterType>3</RequesterType>
        <RequesterAuthenticationID>111112222</RequesterAuthenticationID>
        <RequesterID>AD_THIRDPARTY01</RequesterID>
        <!--Optional:-->
        <RequestPriority>M</RequestPriority>
        <!--Optional:-->
        <CallbackUri>CallBackURI</CallbackUri>
        <CustomerAgreementConfigMessage>
          <MessageHeader>
            <verb>Create</verb>
            <noun>CustomerAgreementConfig</noun>
          </MessageHeader>
          <MessagePayload>
            <CustomerAgreementConfig>
              <SMTRegistered>2</SMTRegistered>
              <CustomerType>1</CustomerType>
              <!--Optional:-->
              <AccountAuthorizationCode>LlcAlpWw</AccountAuthorizationCode>
              <!--Optional:-->
              <SMTEmail/>
              <!--Optional:-->
              <FirstName/>
              <!--Optional:-->
              <LastName/>
              <!--Optional:-->
              <PhoneNumber/>
              <!--Optional:-->
              <LanguagePreference>1</LanguagePreference>
            </CustomerAgreementConfig>
          </MessagePayload>
        </CustomerAgreementConfigMessage>
      </SMTxPHANServicesAgreementRequest>
    </bim:processHANServicesAgreement>
  </soapenv:Body>
</soapenv:Envelope>

```

```
<ServiceSupplier>
  <CompanyName>PGConsulting</CompanyName>
  <!--Optional:-->
  <ContactName/>
  <!--Optional:-->
  <ContactPhone/>
  <!--Optional:-->
  <ContactEmail/>
</ServiceSupplier>
<!--1 to 5 repetitions:-->
<ServiceDeliveryPoint>
  <ESIID>1008901009135635745100</ESIID>
  <EndDeviceList>
    <EndDevice>
      <DeviceMACAddr>001DB7000001D807</DeviceMACAddr>
      <MessageType>1|3</MessageType>
    </EndDevice>
  </EndDeviceList>
</ServiceDeliveryPoint>
<ServiceDeliveryPoint>
  <ESIID>1008901009135635942100</ESIID>
  <EndDeviceList>
    <!--1 to 5 repetitions:-->
    <EndDevice>
      <DeviceMACAddr>001DB7000001D7CF</DeviceMACAddr>
      <MessageType>1|2|3</MessageType>
    </EndDevice>
  </EndDeviceList>
</ServiceDeliveryPoint>
  <!--Optional:-->
  <Comments>HANSERV w/ 2 devices</Comments>
  <Terms>YES</Terms>
</CustomerAgreementConfig>
<MessagePayload>
</CustomerAgreementConfigMessage>
</SMTxPHANServicesAgreementRequest>
</bim:processHANServicesAgreement>
</soapenv:Body>
</soapenv:Envelope>
```

6.3.3.6 In-Home Device Services Agreement Request Acknowledgement

Table 23 shows the Third-Party In-Home Device Services Agreement Request Acknowledgment Schema elements.

Element	Mandatory	Type	Description
RequestID	Y	string(32)	Unique request identifier. This will be generated by SMTxP & sent back in acknowledgement by TDSP
RequestStatus	N	string(3)	Status of the request. Values: ACK - Acknowledgement indicating the request was accepted for further processing. FLR - Failure, indicating the request was not accepted for processing
RequestStatusDesc	N	String(64)	Description of the request status code.
ESIID	Y	string(17,64)	Energy Service Interface Identifier
DeviceMACAddr	N	string(16)	Device MAC Address
Reason	N	string(128)	Reason of Failure at individual ESIID

Table 23: Third-Party In-Home Device Services Agreement Request Acknowledgment Schema Elements

Following XML is a sample of the SOAP message of the In-Home Device Services Agreement Request Acknowledgement:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header/>
  <soapenv:Body>
    <ag:processHANServicesAgreementResponse
      xmlns:ag="http://schemas.esb.ams.com/BIM_SMTxPHANServicesAgreement">
      <SMTxPHANServicesAgreementAck>
        <RequestID>123456</RequestID>
        <AgreementNumber>2807,2808</AgreementNumber>
        <RequestStatus>ACK</RequestStatus>
        <RequestStatusDesc>Success</RequestStatusDesc>
      </SMTxPHANServicesAgreementAck>
    </ag:processHANServicesAgreementResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

6.3.3.7 In-Home Device Agreement Request

Table 24Table 22 shows the Third-Party In-Home Device Agreement Request Schema elements.

Element	Mandatory	Type	Description
SMTRegistered	Y	Int	If SMT = 1 (Unregistered) SMT email address is required; If SMT = 2 (Registered), Account Authorization Code is required
CustomerType	Y	int	Energy Service Interface identifier
AccountAuthorizationCode	N	string(8)	Customer Type: 1 (Residential) or 2 (Business)
SMTEmail	N	string(25)	Provides authorized 3rd Parties the capability to initiate Energy Data Agreement
FirstName	N	string(25)	SMT Email Address
LastName	N	string(25)	If SMT Email Address is provided, SMT will use First Name from customer's profile
PhoneNumber	N	string(10)	If SMT Email Address is provided, SMT will use Last Name from customer's profile
LanguagePreference	N	int	If SMT Email Address is provided, SMT will use Phone Number from customer's profile
CompanyName	Y	string(25)	SMT is the system of record and we use the value available at SMT, when 3rd P inputs a different value.
ContactName	N	string(25)	This should be optional. SMT will use the Contact Name from 3rd Party's company profile if no value was entered for this field
ContactPhone	N	string(10)	This should be optional. SMT will use the Contact Name from 3rd Party's company profile if no value was entered for this field
ContactEmail	N	string(25)	This should be optional. SMT will use the Contact Name from 3rd Party's company profile if no value was entered for this field
ESIID	Y	String (17,64)	Energy Service Interface identifier
DeviceText	N	string(256)	Device Description
DeviceClass	Y	string(5)	Bitmap values. Bit position(right to left) as: 1=InHomeDisplay, 2=LoadControlDevice, 3=ProgrammableThermostat, 4=IntelligentGateway, 5=HomePlugInterface. The SMT API will accept a string(5) value. Characters must be a "0" or a "1". ACK=FLR is returned if this value deviates from these rules.
DeviceMACAddr	Y	string(16)	Device MAC Address
DeviceInstallCode	Y	string(128)	Device Installation Code
DeviceClusterSupport	N	int	Sum of 1 - Messaging, 2 - Load Control, 4 Pricing, 0 - unspecified (max integer value = 7), default=0
Comments	N	string(25)	This is optional.
Terms	Y	string(3)	Terms and Conditions = only accepts value "Yes" - (case insensitive)

Table 24: Third-Party In-Home Device Agreement Request Schema Elements

Following XML is a sample of the SOAP message of the In-Home Device Agreement Request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
    xmlns:inf="http://BIM_SMTxPHANDeviceAgreement_Library/INF_SMTxPHANDeviceAgreement">  
    <soapenv:Header/>  
    <soapenv:Body>  
        <inf:processHANDeviceAgreement>  
            <SMTxPHANDeviceAgreementRequest>  
                <RequesterType>3</RequesterType>  
                <RequesterAuthenticationID>119222333</RequesterAuthenticationID>  
                <RequesterID>CJ_THIRDPARTY9_A1</RequesterID>  
                <!--Optional:-->  
                <RequestPriority>M</RequestPriority>  
                <!--Optional:-->  
                <CallbackUri>CallBackURI</CallbackUri>  
                <!--Optional:-->  
                <CustomerAgreementConfigMessage>  
                    <MessageHeader>  
                        <verb>Create</verb>  
                        <noun>CustomerAgreementConfig</noun>  
                    </MessageHeader>  
                    <MessagePayload>  
                        <CustomerAgreementConfig>  
                            <SMTRegistered>1</SMTRegistered>  
                            <CustomerType>2</CustomerType>  
                            <!--Optional:-->  
                            <AccountAuthorizationCode></AccountAuthorizationCode>  
                            <!--Optional:-->  
                            <SMTEmail>ss@co.in</SMTEmail>  
                            <!--Optional:-->  
                            <FirstName>Fname</FirstName>  
                            <!--Optional:-->  
                            <LastName>Ls name</LastName>  
                            <!--Optional:-->  
                            <PhoneNumber>?</PhoneNumber>  
                            <!--Optional:-->  
                            <LanguagePreference>1</LanguagePreference>  
                            <ServiceSupplier>  
                                <CompanyName>Comp name</CompanyName>  
                            </ServiceSupplier>  
                            <!--Zero or more repetitions:-->  
                            <ServiceDeliveryPoint>  
                                <ESIID>1008901020194118435100</ESIID>  
                                <!--Optional:-->  
                            <EndDeviceList>
```

```
<EndDevice>
<DeviceText>First</DeviceText>
<DeviceClass>00001</DeviceClass>
<!--Optional:-->
<DeviceMACAddr>001DB7000001D7CA</DeviceMACAddr>
<DeviceInstallCode>kuchbhidalchalega</DeviceInstallCode>
<!--Optional:-->
</EndDevice>
</EndDeviceList>
</ServiceDeliveryPoint>
<ServiceDeliveryPoint>
<ESIID>1008901020194118435100</ESIID>
<!--Optional:-->
<EndDeviceList>
<EndDevice>
<DeviceText>Second</DeviceText>
<DeviceClass>00010</DeviceClass>
<!--Optional:-->
<DeviceMACAddr>001DB7000001D807</DeviceMACAddr>
<DeviceInstallCode>0962BEE35014A40F3E72</DeviceInstallCode>
<!--Optional:-->
</EndDevice>
</EndDeviceList>
</ServiceDeliveryPoint>
<ServiceDeliveryPoint>
<ESIID>1008901020194118435100</ESIID>
<!--Optional:-->
<EndDeviceList>
<EndDevice>
<DeviceText>Third</DeviceText>
<DeviceClass>00100</DeviceClass>
<!--Optional:-->
<DeviceMACAddr>001DB7000001D819</DeviceMACAddr>
<DeviceInstallCode>7E59661334ED0F5F4CDF</DeviceInstallCode>
<!--Optional:-->
</EndDevice>
</EndDeviceList>
</ServiceDeliveryPoint>
<ServiceDeliveryPoint>
<ESIID>1008901020194118435100</ESIID>
<!--Optional:-->
<EndDeviceList>
<EndDevice>
<DeviceText>Fourth</DeviceText>
```

```

<DeviceClass>01000</DeviceClass>
<!--Optional:-->
<DeviceMACAddr>001DB7000001D80E</DeviceMACAddr>
<DeviceInstallCode>EF8723DD773648ACC12E</DeviceInstallCode>
<!--Optional:-->
</EndDevice>
</EndDeviceList>
</ServiceDeliveryPoint>
<ServiceDeliveryPoint>
<ESIID>1008901020194118435100</ESIID>
<!--Optional:-->
<EndDeviceList>
<EndDevice>
<DeviceText>Fifth</DeviceText>
<DeviceClass>10000</DeviceClass>
<!--Optional:-->
<DeviceMACAddr>001DB7000001D7C5</DeviceMACAddr>
<DeviceInstallCode>61ED4A747E6470EE0112</DeviceInstallCode>
<!--Optional:-->
</EndDevice>
</EndDeviceList>
</ServiceDeliveryPoint>
<!--Optional:-->
<ValidityInterval/>
<!--Optional:-->
<Comments>Group 7</Comments>
<Terms>Yes</Terms>
</CustomerAgreementConfig>
</MessagePayload>
</CustomerAgreementConfigMessage>
</SMTxPHANDeviceAgreementRequest>
</inf:processHANDeviceAgreement>
</soapenv:Body>
</soapenv:Envelope>

```

6.3.3.8 In-Home Device Agreement Request Acknowledgement

Table 25 shows the Third-Party In-Home Device Agreement Request Acknowledgment Schema elements.

Element	Mandatory	Type	Description
RequestID	Y	string(32)	Unique request identifier. This will be generated by SMTxP & sent back in acknowledgement by TDSP

Element	Mandatory	Type	Description
RequestStatus	N	string(3)	Status of the request. Values: ACK - Acknowledgement indicating the request was accepted for further processing. FLR - Failure, indicating the request was not accepted for processing
RequestStatusDesc	N	String(64)	Description of the request status code.
ESIID	Y	string(17,64)	Energy Service Interface Identifier
DeviceMACAddr	N	string(16)	Device MAC Address
Reason	N	string(128)	Reason of Failure at individual ESIID

Table 25: Third-Party In-Home Device Agreement Request Acknowledgement Schema Elements

Following XML is a sample of the SOAP message of the In-Home Device Agreement Request Acknowledgement:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header/>
  <soapenv:Body>
    <ag:processHANDeviceAgreementResponse
      xmlns:ag="http://BIM_SMTxPHANDeviceAgreement_Library/INF_SMTxPHANDeviceAgreement">
      <SMTxPHANDeviceAgreementAck>
        <RequestID>1234</RequestID>
        <AgreementNumber>3404,3405,3406,3407,3408</AgreementNumber>
        <RequestStatus>ACK</RequestStatus>
        <RequestStatusDesc>This ESIID belongs to a different NON-RES user.</RequestStatusDesc>
      </SMTxPHANDeviceAgreementAck>
    </ag:processHANDeviceAgreementResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

6.3.3.9 Re-Add In-Home Device Request

Table 26 shows the Third-Party Re-add In-Home Device Request Schema elements.

Element	Mandatory	Type	Description
AgreementNumber	Y	string(12)	The Energy Data Agreement Number.
ESIID	Y	string(17,64)	Energy Service Interface Identifier
DeviceMACAddress	Y	string(16)	Device MAC Address

Table 26: Third-Party Re-Add In-Home Device Request Schema Elements

Following XML is a sample of the SOAP message of the Re-add In-Home Device Request:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:inf="http://BIM_SMTxPHANDeviceAgreement_Library/INF_SMTxPHANDeviceAgreement">
  <soapenv:Header/>
  <soapenv:Body>
    <inf:processHANDeviceReAdd>
      <SMTxPHANDeviceReAddRequest>
        <!--Optional:-->
        <RequestID/>
        <RequesterType>3</RequesterType>
        <RequesterAuthenticationID>121222333</RequesterAuthenticationID>
        <RequesterID>CJ_REP3_A1</RequesterID>
        <!--Optional:-->
        <CallbackUri/>
        <RequestPriority>M</RequestPriority>
        <CustomerAgreementConfigMessage>
          <MessageHeader>
            <verb>Create</verb>
            <noun>CustomerAgreementConfig</noun>
          </MessageHeader>
          <MessagePayload>
            <AgreementConfig>
              <AgreementNumber>1737</AgreementNumber>
              <ESIID>1008901017189815969100</ESIID>
            </AgreementConfig>
            <EndDeviceList>
              <EndDevice>
                <DeviceMacAddr>002446000006734E</DeviceMacAddr>
              </EndDevice>
            </EndDeviceList>
            </MessagePayload>
          </CustomerAgreementConfigMessage>
        </SMTxPHANDeviceReAddRequest>
      </inf:processHANDeviceReAdd>
    </soapenv:Body>
  </soapenv:Envelope>

```

6.3.3.10 Re-Add In-Home Device Request Acknowledgement

Table 27Table 25 shows the Third-Party In-Home Device Agreement Request Acknowledgment Schema elements.

Element	Mandatory	Type	Description
---------	-----------	------	-------------

Element	Mandatory	Type	Description
RequestID	Y	string(32)	Unique request identifier. This will be generated by SMTxP & sent back in acknowledgement by TDSP
RequestStatus	N	string(3)	Status of the request. Values: ACK - Acknowledgement indicating the request was accepted for further processing. FLR - Failure, indicating the request was not accepted for processing
RequestStatusDesc	N	String(64)	Description of the request status code.
EIID	Y	string(17,64)	Energy Service Interface Identifier
DeviceMACAddr	N	string(16)	Device MAC Address
Reason	N	string(128)	Reason of Failure at individual EIID

Table 27: Third-Party In-Home Device Agreement Request Acknowledgement Schema Elements

Following XML is a sample of the SOAP message of the Re-add In-Home Device Request Acknowledgement:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header/>
  <soapenv:Body>
    <ag:processHANDeviceReAddResponse
      xmlns:ag="http://BIM_SMTxPHANDeviceAgreement_Library/INF_SMTxPHANDeviceAgreement">
      <SMTxPHANDeviceReAddAck>
        <RequestID></RequestID>
        <RequestStatus>ACK</RequestStatus>
        <RequestStatusDesc>Success</RequestStatusDesc>
      </SMTxPHANDeviceReAddAck>
    </ag:processHANDeviceReAddResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

6.3.3.11 Scheduled Usage Report Request

Table 28 shows the Third-Party Scheduled Usage Report Request Schema elements.

Element	Mandatory	Type	Description
---------	-----------	------	-------------

Element	Mandatory	Type	Description
ReportType	Y	string(1)	Type of report/response. Accepted values are: 1: INTERVAL - interval usage report from startDate to endDate 2: DAILY - daily usage report generated from startDate to endDate 3: MONTHLY - monthly usage report from startDate to endDate
StartDate	Y	string(10)	Start date of when 3rd Party will receive the report
EndDate	Y	string(10)	End date of when 3rd party will be stopped receiving report
ESIID	Y	string(17,64)	Energy Service Interface Identifier

Table 28: Third-Party Scheduled Usage Report Request Schema Elements

Following XML is a sample of the SOAP message of the Scheduled Usage Report Request:

```

<soapenv:Envelope xmlns:bim="http://schemas.esb.ams.com/BIM_BIM_SMTxPScheduledReport"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body wsu:Id="id-19601805" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
  200401-wss-wssecurity-utility-1.0.xsd">
    <bim:processScheduledReport>
      <SMTxPScheduledReportRequest>
        <!--Optional:-->
        <RequestID>9865498</RequestID>
        <RequesterType>3</RequesterType>
        <RequesterAuthenticationID>3234234569685</RequesterAuthenticationID>
        <RequesterID>THIRDPARTYADM151</RequesterID>
        <RequestPriority>M</RequestPriority>
        <!--Optional:-->
        <CallbackUri>CallbackURI</CallbackUri>
        <MeterUsageReqList>
          <!--1 to 3 repetitions:-->
          <ReportTypeArray>
            <ReportType>1</ReportType>
            <StartDate>10/29/2014</StartDate>
            <EndDate>11/13/2014</EndDate>
          </ReportTypeArray>
          <AgreementNumber>1385</AgreementNumber>
          <ESIID>10409514678899999</ESIID>
        </MeterUsageReqList>
      </SMTxPScheduledReportRequest>
    </bim:processScheduledReport>
  </soapenv:Body>
</soapenv:Envelope>
```

6.3.3.12 Scheduled Usage Report Request Acknowledgement

Table 29 Table 25 shows the Third-Party Scheduled Usage Report Request Acknowledgment Schema elements.

Element	Mandatory	Type	Description
RequestID	Y	string(32)	Unique request identifier. This will be generated by SMTxP & sent back in acknowledgement by TDSP
RequestStatus	N	string(3)	Status of the request. Values: ACK - Acknowledgement indicating the request was accepted for further processing. FLR - Failure, indicating the request was not accepted for processing
RequestStatusDesc	N	String(64)	Description of the request status code.
ESIID	Y	string(17,64)	Energy Service Interface Identifier
Reason	N	string(128)	Reason of Failure at individual ESIID

Table 29: Third-Party Scheduled Usage Report Request Acknowledgement Schema Elements

Following XML is a sample of the SOAP message of the Scheduled Usage Report Request Acknowledgement:

```

<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <re:processScheduledReportResponse
      xmlns:re="http://schemas.esb.ams.com/BIM_BIM_SMTxPScheduledReport">
      <SMTxPScheduledReportAck>
        <RequestID>967523</RequestID>
        <OrderID>oLCRbdMkIJCSrRjK</OrderID>
        <RequestStatus>ACK</RequestStatus>
        <RequestStatusDesc>Success</RequestStatusDesc>
      </SMTxPScheduledReportAck>
    </re:processScheduledReportResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

7. Ad-hoc Usage soapUI Starter Kit

7.1 Overview of the soapUI Starter Kit for Ad-hoc Usage Queries

This section describes a soapUI Starter Kit that REPSs and Third-Parties can use for invoking the Smart Meter Texas (SMT) Ad-Hoc Query Web Service interface.

7.1.1 Intended Audience

Users of this Starter Kit are assumed to have basic technical expertise in these areas:

- Downloading applications from the Internet
- Installation and setup of applications
- Creation and editing of XML documents
- Internet security including user ids, passwords, and the use of X.500 certificates
- Network communications including URLs, IP addresses and common network troubleshooting commands such as ping, telnet, FTP, etc.

7.1.2 About soapUI

soapUI is a free, open source desktop application available for download via the WWW (visit <http://www.soapUI.org>) for inspecting and invoking Web Services.

7.1.3 About this Starter Kit

This Starter Kit comes with the following:

- Ad-hoc query interface document – describes interface functions
- soapUI project – contains XML interface definitions that can be used by the soapUI application
- SMT's UAT/Staging SSL Certificate

Prerequisites and steps for using the Starter Kit are described below.

7.2 Prerequisites

This section presents the prerequisites for using the Ad-hoc Usage Query Starter Kit.

7.2.1 Environmental Setup

Prior to using this Starter Kit, Third-Parties and REPs must provide the following Staging and/or Production information.

Staging

REPs and Third-Parties can submit this environmental information for access to the SMT Staging environment:

- SSL certification – SMT will accept a Certificate Authority (CA) or self-signed certificate. A Third-Party or REP needs to provide to SMT with their SSL certificate for Staging environment setup.
- Signing certificate – SMT will accept a CA or self-signed signer certificate. A REP or Third-Party will need to provide SMT with their signer certificate for Staging environment setup.
- DUNS – An SMT system account will be created that the REP or Third-Party must use when accessing the SMT Staging environment. One DUNS number must be associated with the SMT system account. REP may have more than one DUNS number associated with the same system account. Only one DUNS number is allowed for Third-Party.
- Entity Name – This is the preferred Company name.

Third-Parties and REPs can submit this information via email to support@smartmetertexas.com.

A REP or Third-Party needs SMT's SSL certificate for the Staging to enable mutual authentication. The SMT certificate is included as part of this Starter Kit.

Once SMT has setup the system account for Staging, account information will forwarded to the Third-Party or REP's organization.

Production

REPs and Third-Parties can submit this environmental information for access to the SMT Production environment.

- SSL certification – SMT will only accept a CA certificate. A Third-Party or REP needs to provide to SMT with their CA SSL certificate for Production environment setup.
- Signing certificate – SMT will accept a CA signer certificate. A Third-Party or REP will need to provide SMT with their CA signer certificate for Production environment setup.
- DUNS – An SMT system account will be created that REPs must use when accessing the SMT Staging environment. One or more DUNS number must be associated with the SMT system account for REP. Only one DUNS number is allowed for Third-Party.
- IP Address(es) – Access is restricted to the SMT Production Environment. The Third-Party or REP's IP Address is needed to enable access.
- Entity Name – This is the preferred Company name.

Third-Parties and REPs can submit this information via email to support@smartmetertexas.com.

SMT uses a CA certificate for SSL in Production. SMT's certificate will be downloaded automatically via soapUI.

Once SMT has setup the system account for Production, account information will be forwarded to the Third-Party or REP's organization.

7.2.2 Key Store

An SSL KeyStore is required to invoke Web Services as part of the SMT starter kit.

The Keystore that is supported on SoapUI is JKS (Java Key Store). JKS is supported by¹:

- iKeyMan
- Keytool and
- OpenSSL.

Staging

Self-signed or CA certificates may be used in Staging. When using self-signed certificates, follow these steps:

1. Create a JKS (Java Key Store).
2. Create Self signed SSL certificate.
3. Create Self signed signer certificate.
4. After the self signed SSL Certificate and Signer Certificates are generated, provide these two certificates to SMT as discussed in Section 7.2.1 under Staging.
5. Insert the self-signed certificates into the JKS if they are not put there during the creation process.

CA-issued certificates can be used in Staging and Production so are preferred to self-signed certificates. When using self-signed certificates, follow these steps:

1. Obtain certificates.
 2. Create a JKS (Java Key Store).
 3. Provide these two certificates to SMT as discussed in Section 7.2.1 under Staging.
 4. Inserting the CA certificates into the JKS.
 5. Insert the SMT Staging certificate into the JKS that is included with this Starter Kit.
-

¹ iKeyMan is GUI tool for Key Management that comes as part of WAS(Websphere Application Server) and is available on <WAS_Home>/bin. One can launch iKeyMan on Dos prompt by running the command WAS_Home>/bin/iKeyMan. If WAS is not available, one has to download iKeyMan and associated jar files from Websphere Info center. If iKeyMan is used as stand alone component, the configuration steps need to be followed. Java Keytool comes as part of standard JDK. One can download JDK from <http://www.java.com>. OpenSSL is available on www.openssl.org.

Production

Only CA-issued certificate will be accepted in Production. Perform following steps:

1. Obtain certificates.
2. Create a JKS (Java Key Store).
3. Provide these two certificates to SMT as discussed in Section 7.2.1 under Staging.
4. Inserting the CA certificates into the JKS.

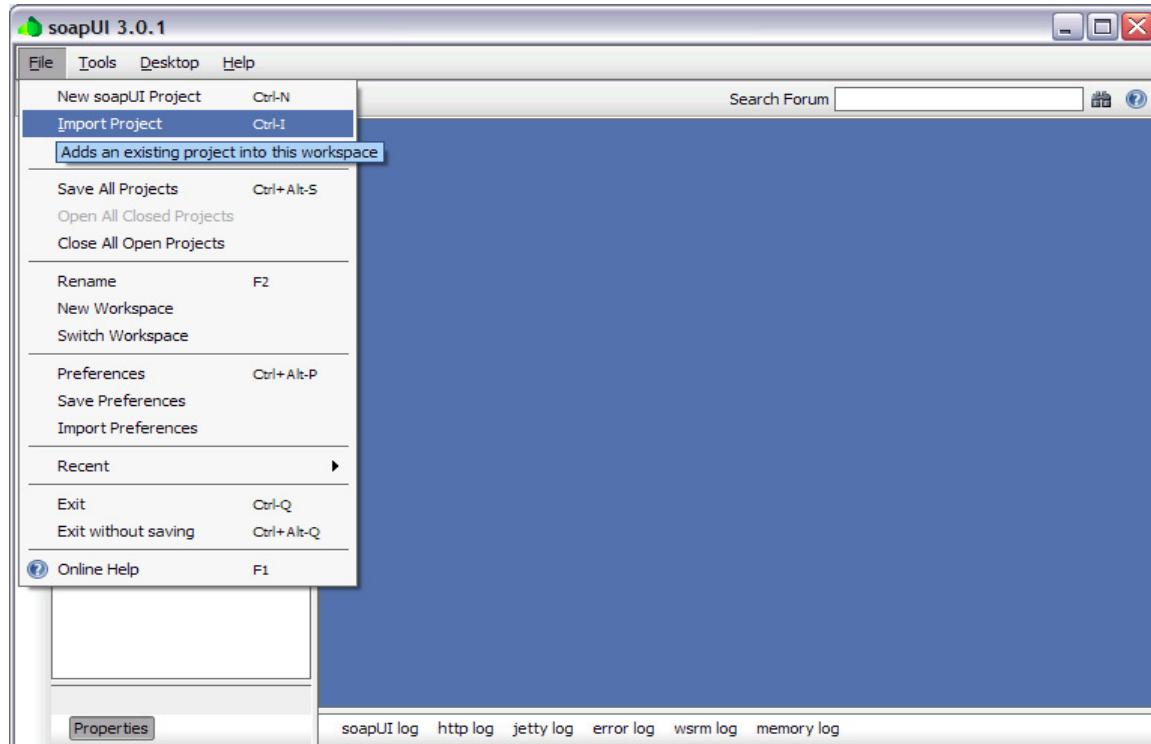
7.3 Overview of Steps for using the Starter Kit

The following sections describe how to:

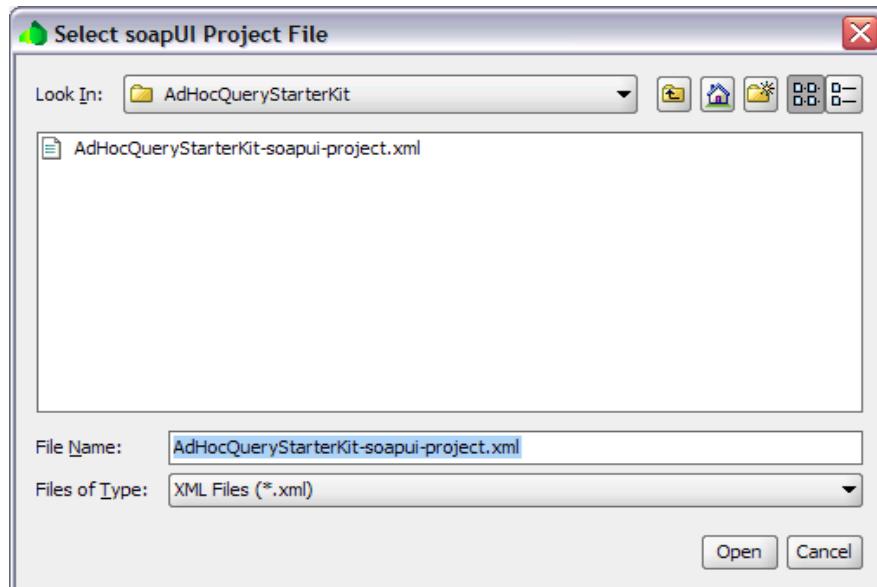
1. Set up the Starter Kit
2. Configure SSL
3. Configure WS-Security
4. Invoke Web Services

7.4 Setting up the Starter Kit

Place the Starter Kit files in a target directory. Then, Launch soapUI and click on File -> Import Project as shown below.



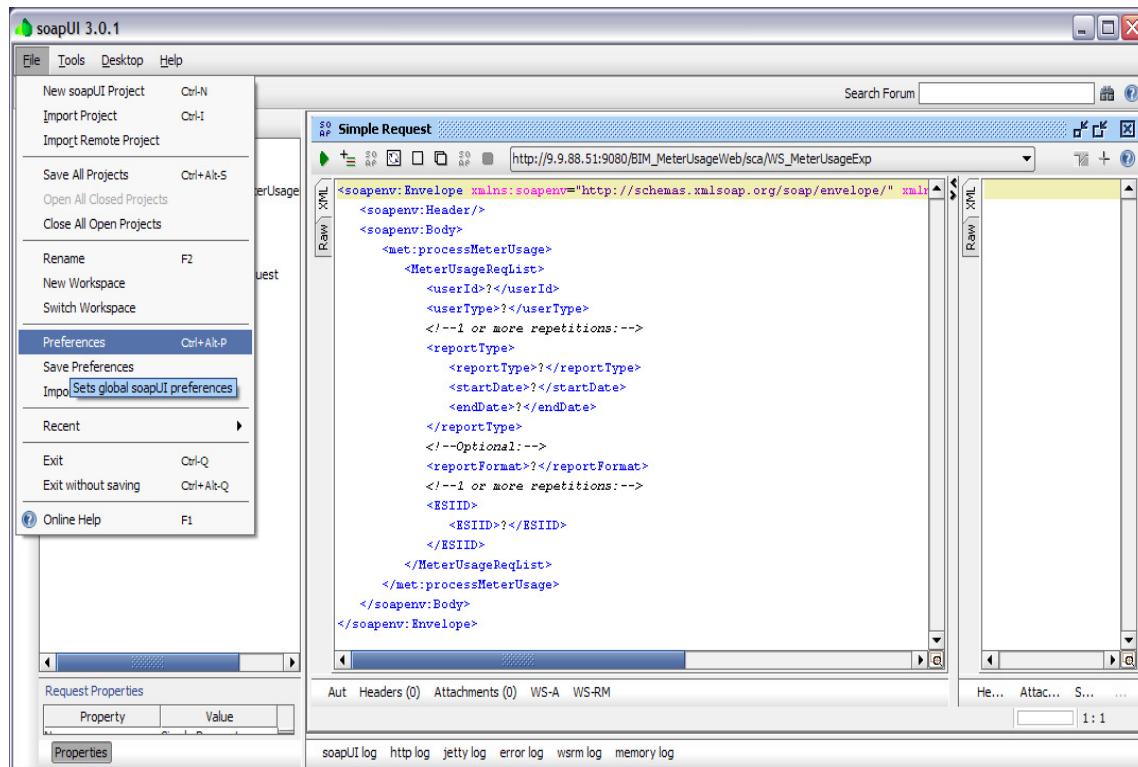
Navigate to directory and the Starter Kit project file and select it. Click Open. This is pictured below.



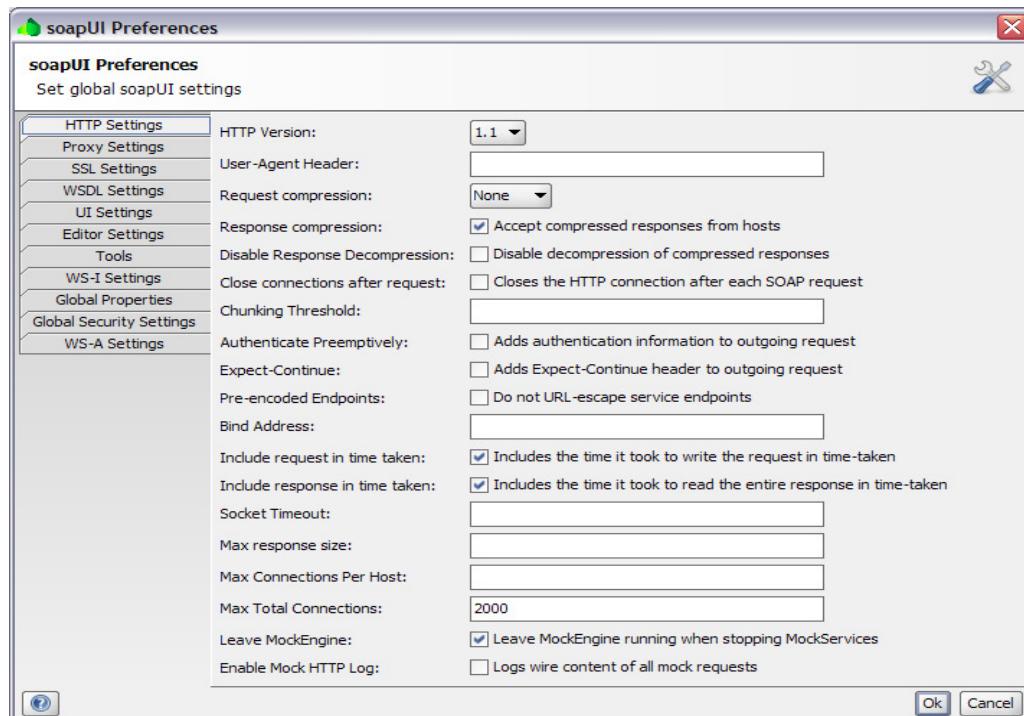
soapUI will load the Starter Kit project.

7.5 SSL Configuration

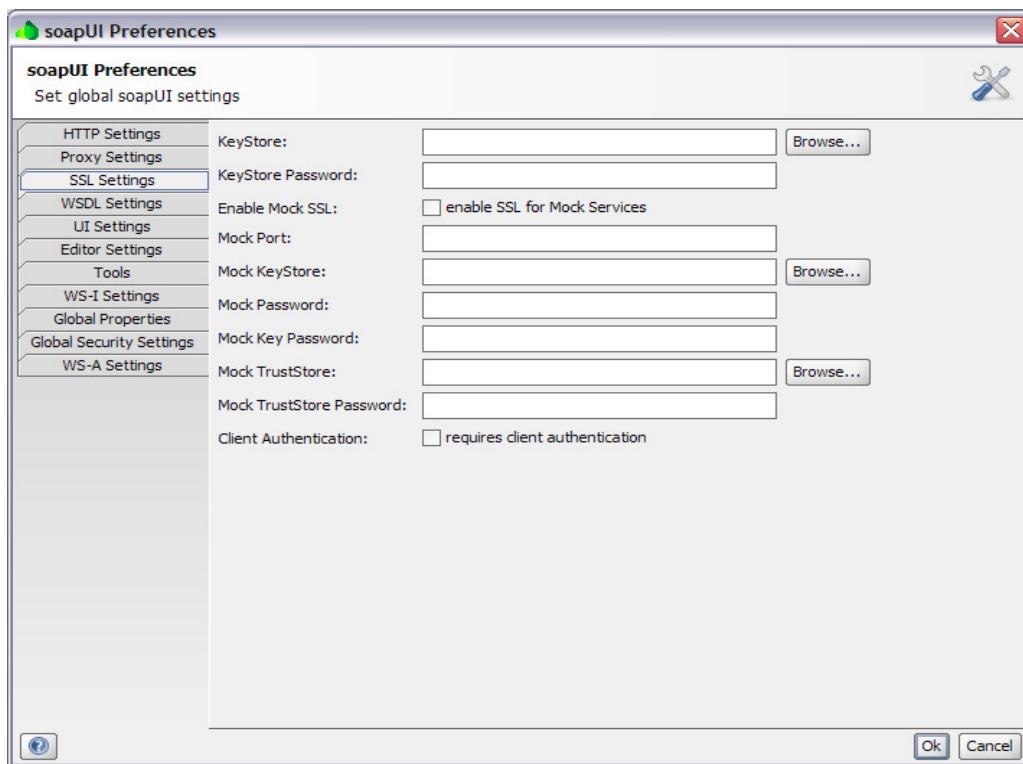
Launch SoapUI and click on File -> Preferences as shown below.



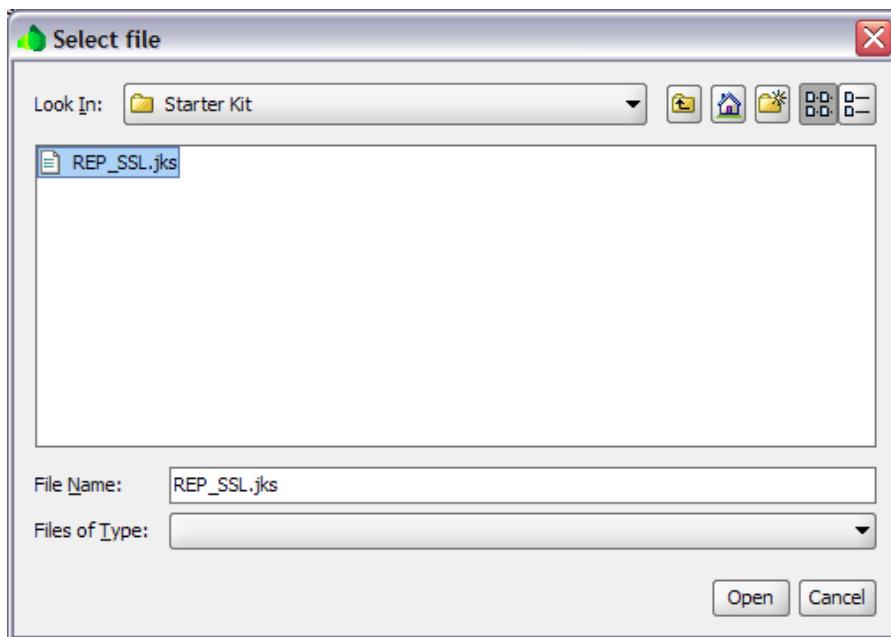
The soapUI Preferences will pop-up in a separate window.



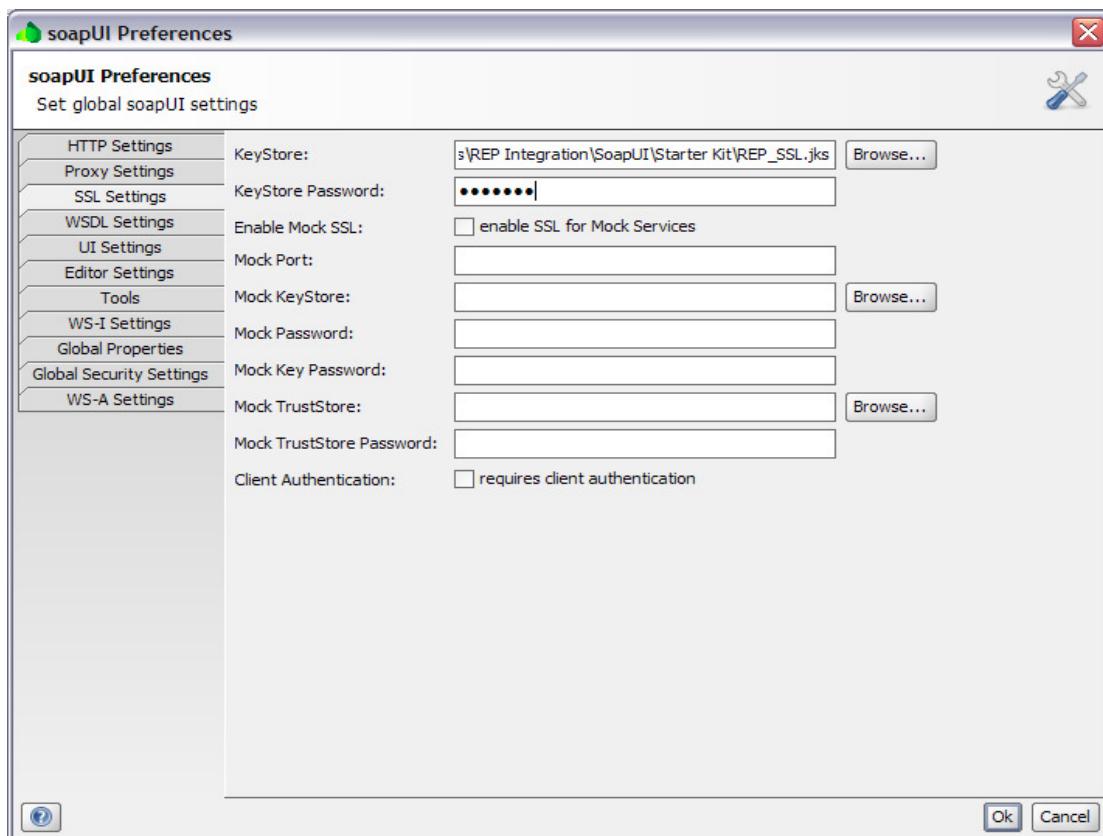
Select SSL Settings on the left navigation bar.



Click on browse button in Keystore field to select Keystore to use for SSL.



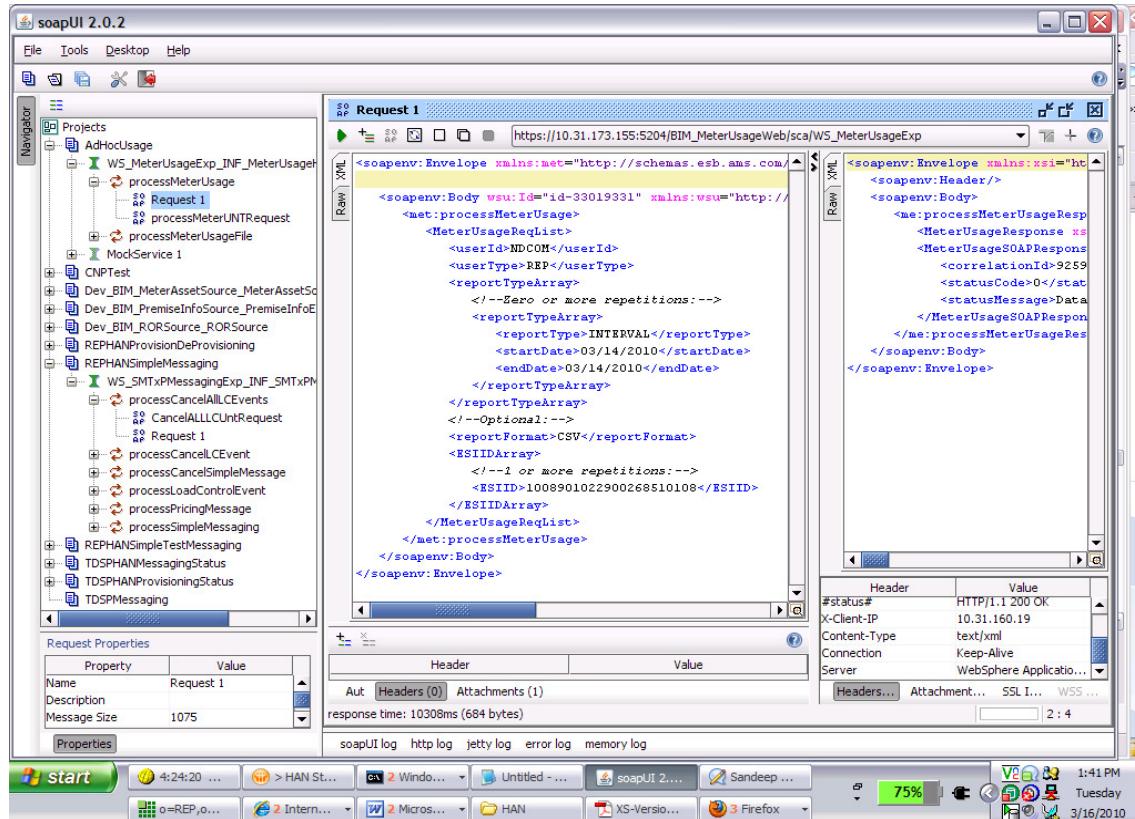
Input your Keystore password, for example, SMT1234, and click Ok.



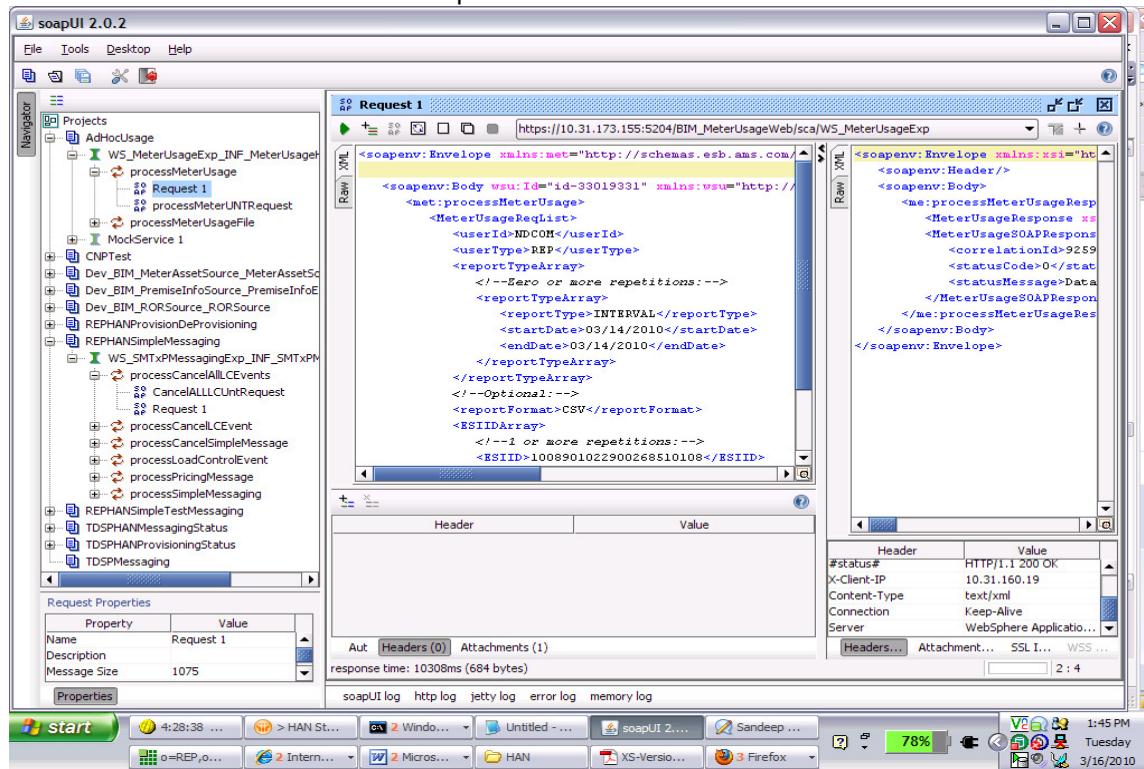
7.6 HTTP Header Configuration

An HTTP header with the name of ENTITY_NAME must be added to Ad-hoc query requests. The HTTP header needs to contain a value that is the entity name (Third-Party or REP) that is submitting the request.

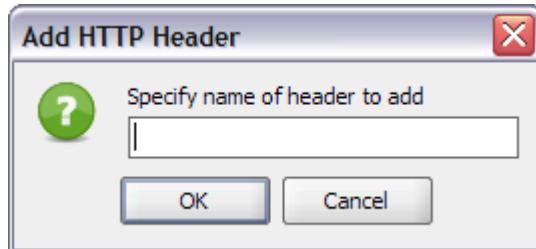
To add a HTTP header, locate the Headers field at the bottom middle of a request window as depicted below.



Double click on the Headers field to open the Headers information box.

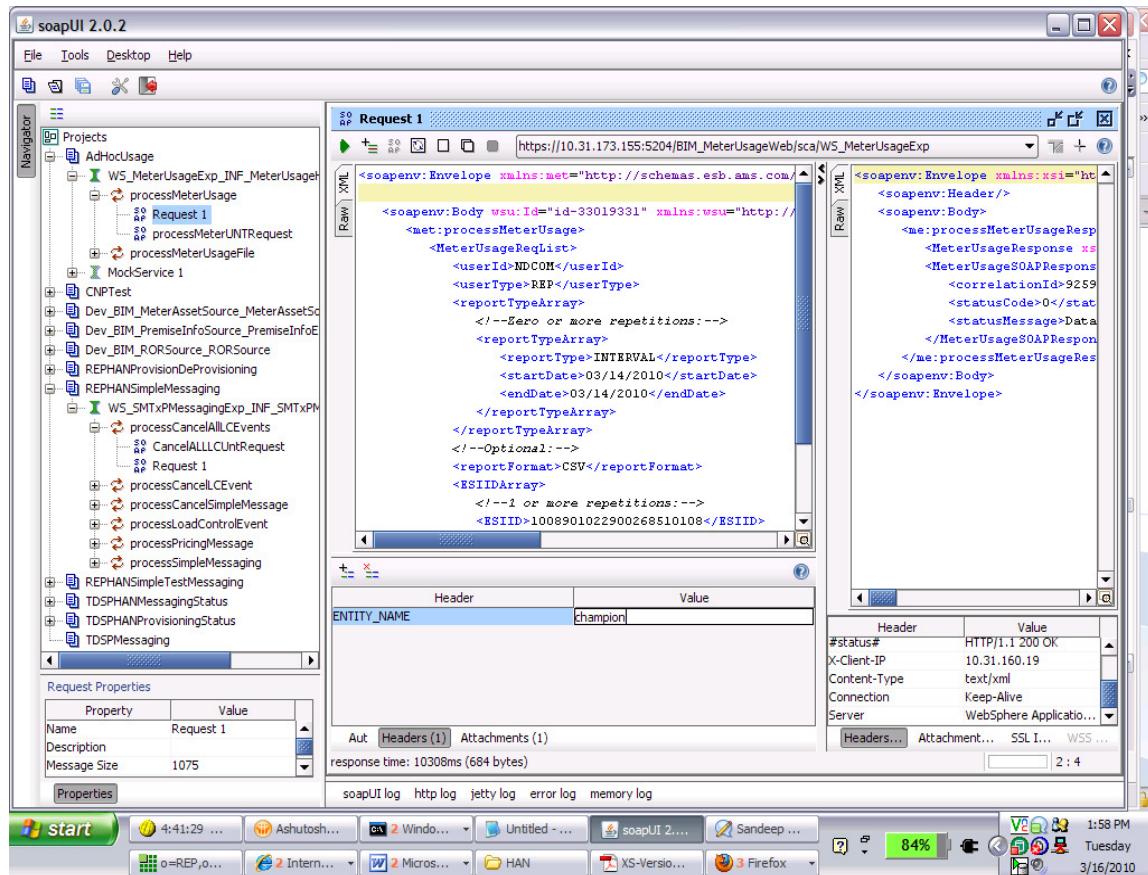


Clicking the + sign above the Headers information box to obtain a HTTP headers name input box as seen below.



Enter ENTITY_NAME and click OK.

An HTTP header with a name of ENTITY_NAME had been added to the request as seen below. In the value field, enter entity name or string that is recognized by SMT as the entity. The value should be the Entity Name provided as part of environment set up – see Section 7.2.1.

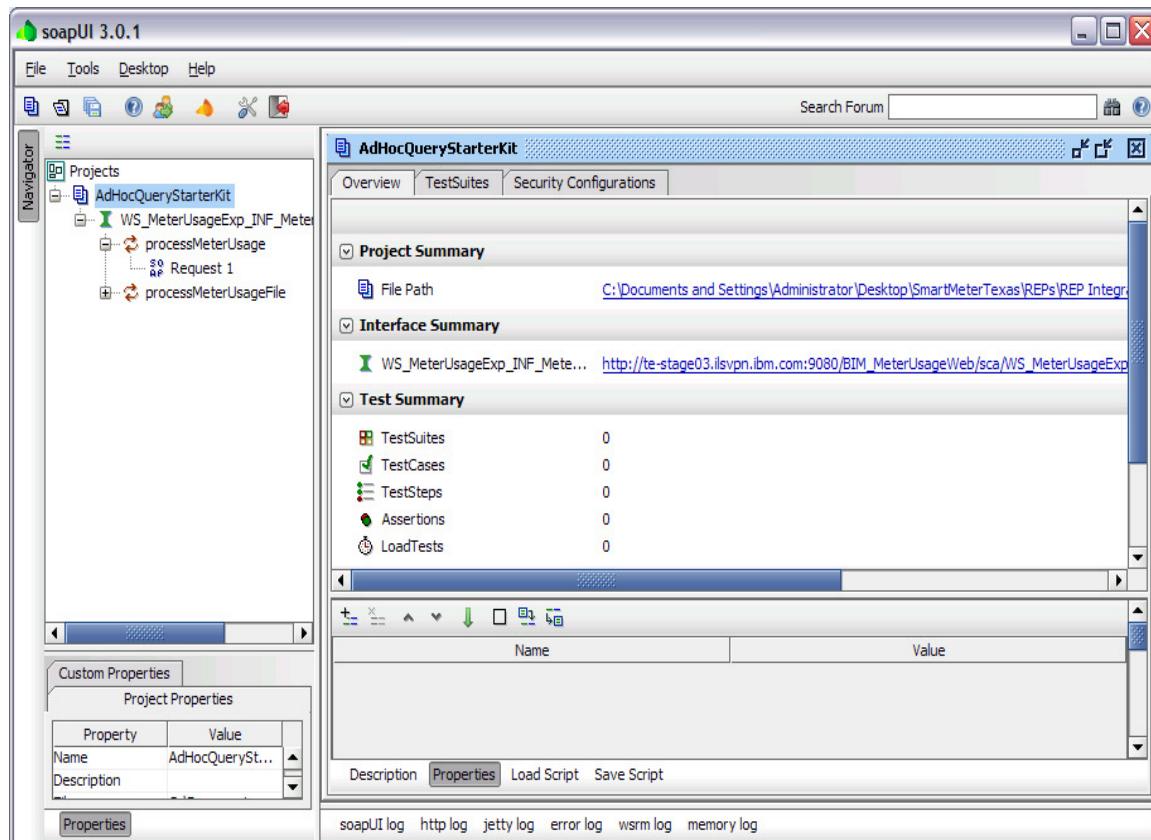


The above configuration allows SMT to determine source or origination of the request.

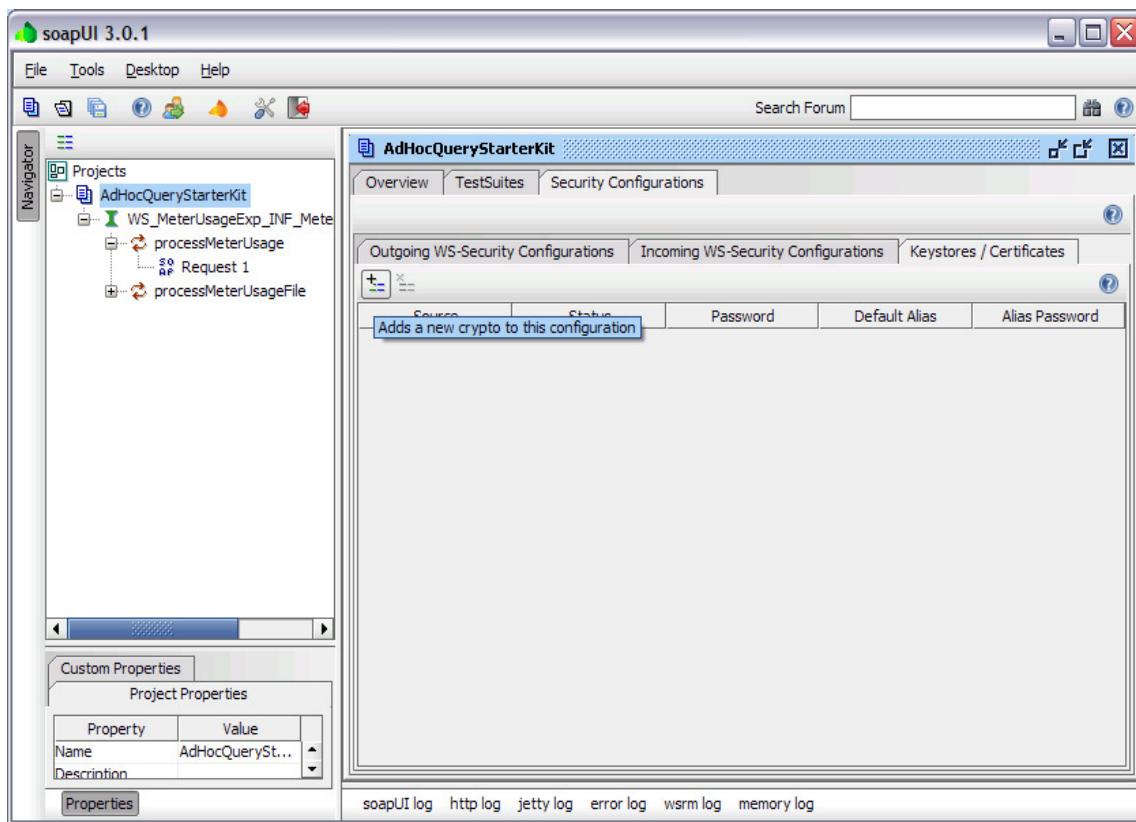
7.7 WS-Security Configuration

This section explains how to configure a WS-Security signature for outgoing Web Services call. The Time Stamp, UserNameToken and SOAP Body are signed. soapUI must be configured for Web Service calls.

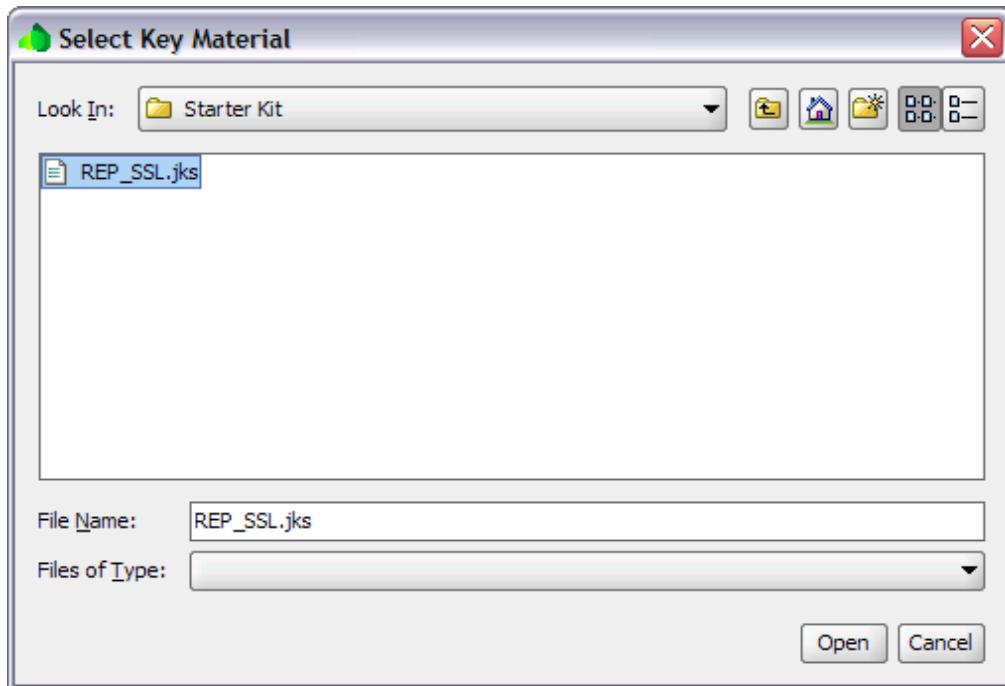
Select soapUI project and double click.



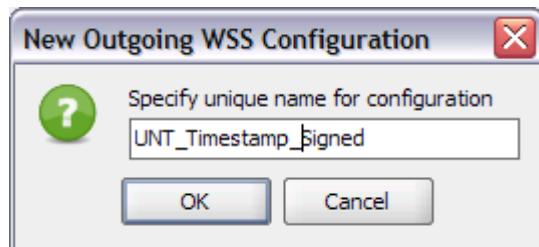
Select the Security Configurations tab and then the Keystore/Certificates tab as pictured below.



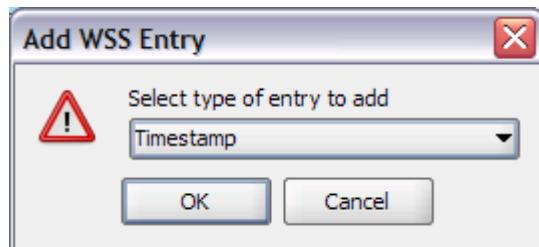
Click on + sign to specify Keystore/Certificate.



Now, click on Outgoing WS-Security Configuration tab. Then click on + sign and enter a unique configuration name. Below, UNT_Timestamp_Signed is used to represent: user name token, timestamp, and signed.



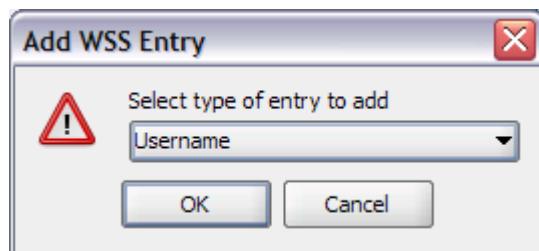
In the lower half panel. Click on + sign and select Timestamp as type of entry to add.



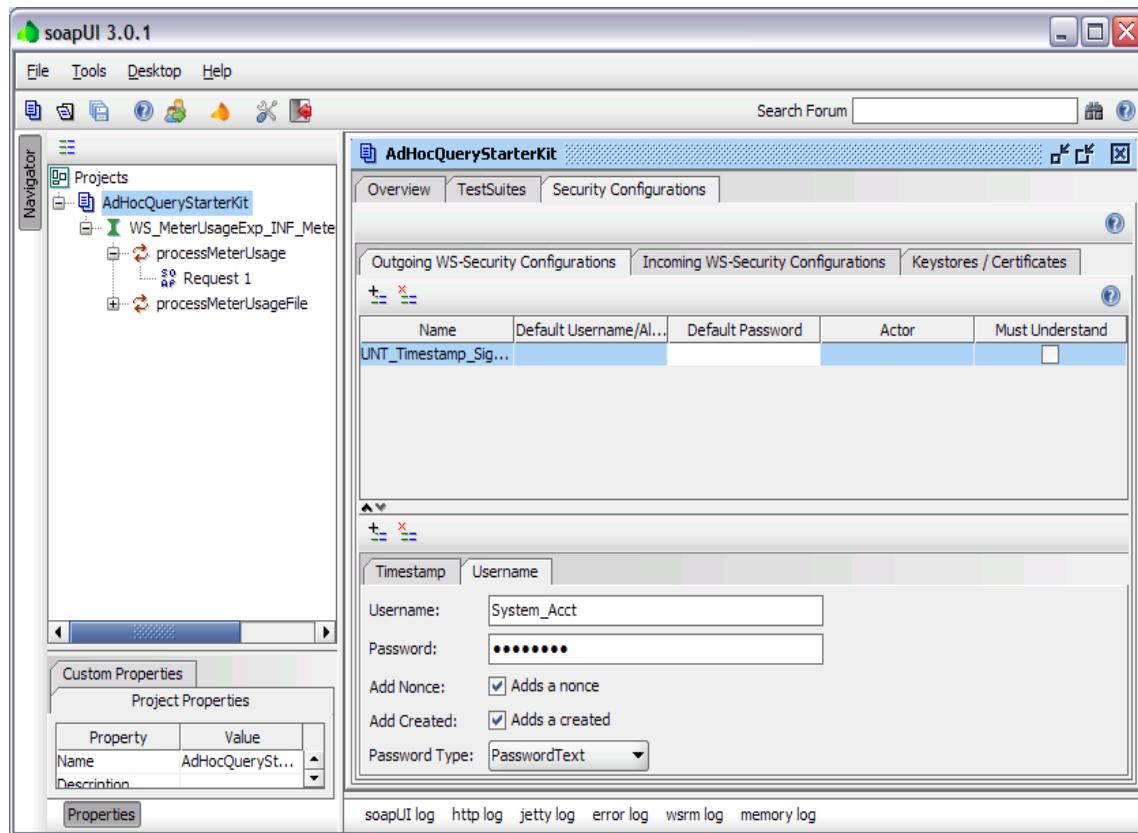
Enter Time to live value, for example, 60.

The screenshot shows the soapUI 3.0.1 interface with the 'AdHocQueryStarterKit' project selected in the Navigator. The 'Security Configurations' tab is open, specifically the 'Outgoing WS-Security Configurations' section. A table lists a configuration named 'UNT_Timestamp_Signed'. Below the table, a 'Timestamp' configuration panel is displayed, showing 'Time To Live:' set to 60 and 'Millisecond Precision:' checked. The status bar at the bottom indicates logs for soapUI log, http log, jetty log, error log, wsrm log, and memory log.

Select + again in lower half panel and select Username as type of entry to add.



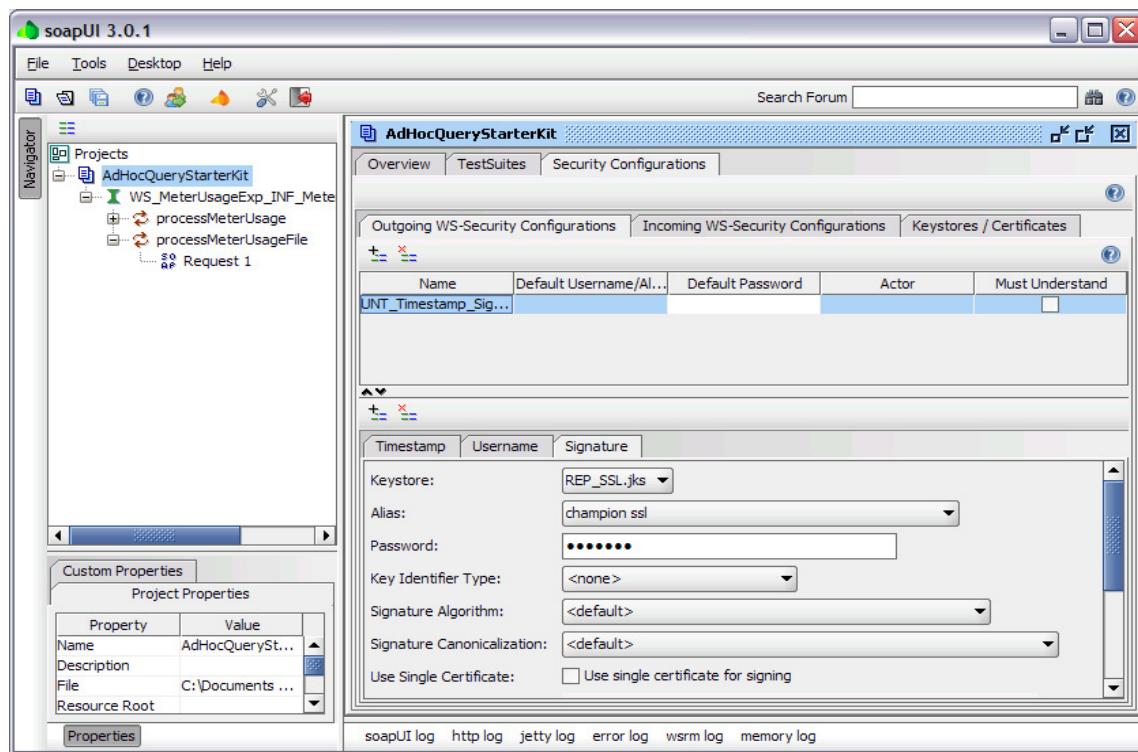
Specify Username, password and PasswordText as type of password to add. (Specifying password is mandatory).



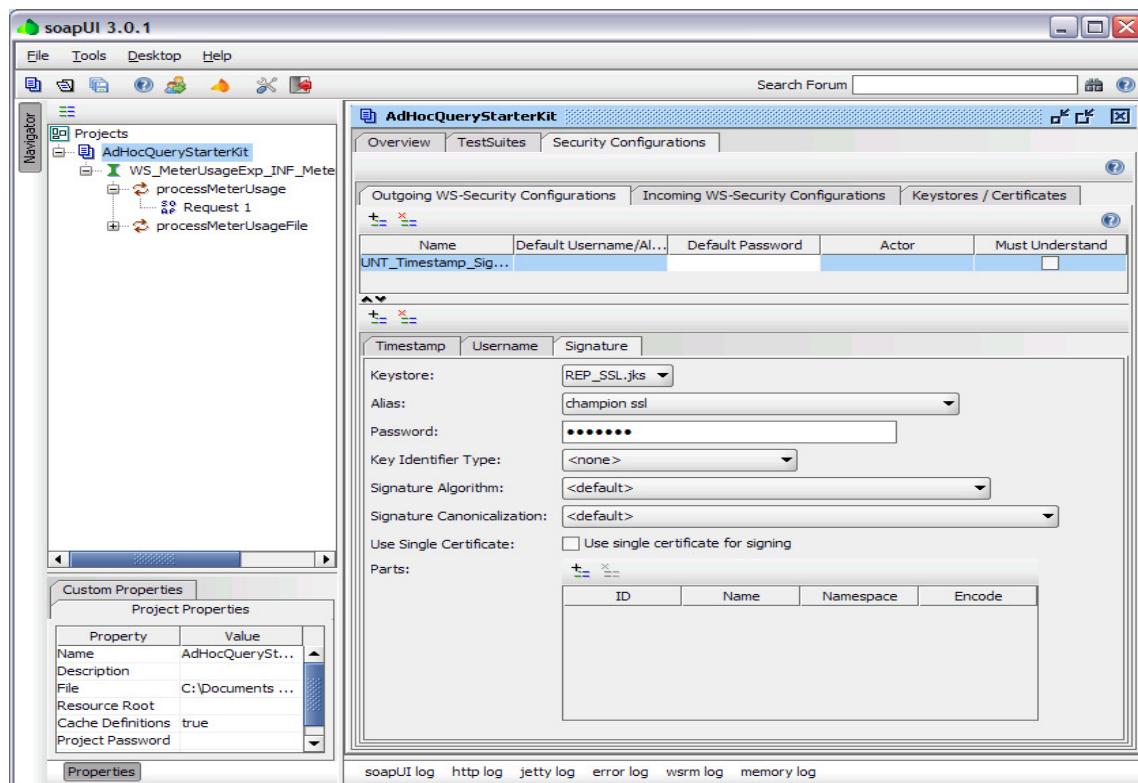
Now, select + sign at the lower half panel and specify Signature as type of entry to add.



Specify Keystore, Alias and Keystore Password.



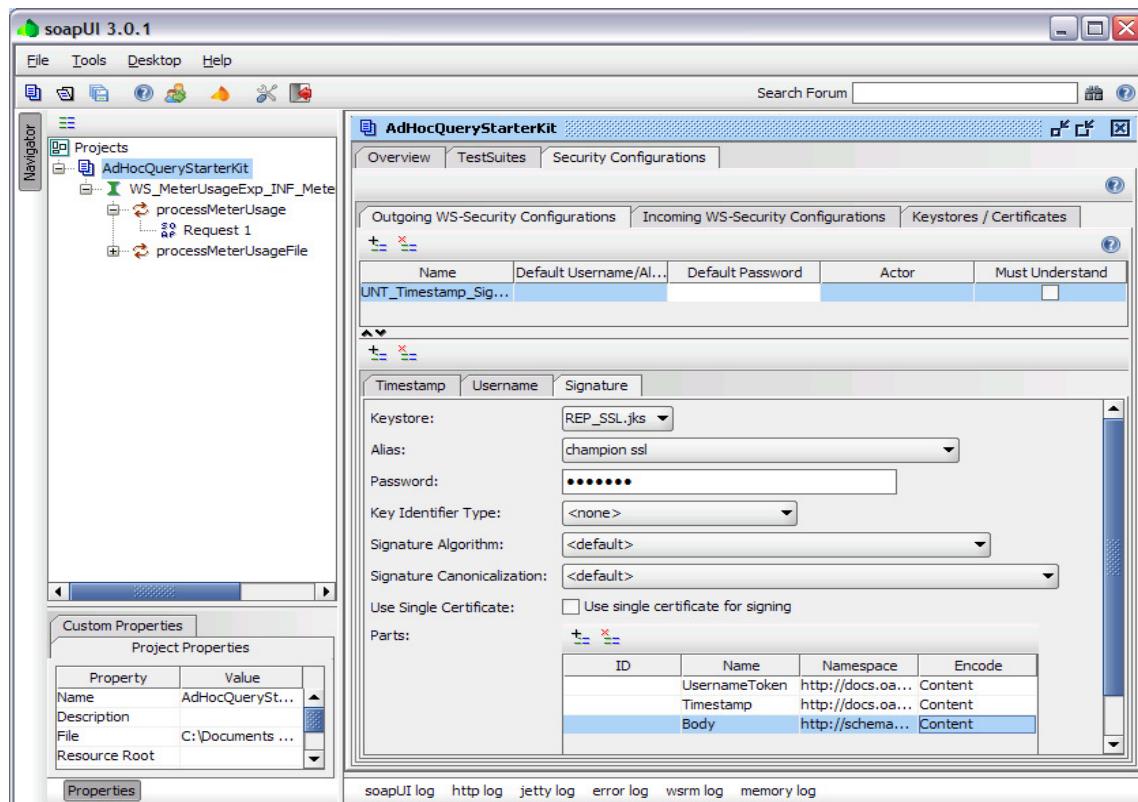
Then, select the correct Binary Security Token in Key Identifier Type field.



Now, enter the different parts of the signature. Click Parts + after the *Use Single Certificate* field to enter the information in the following table. (Click the + 3 times for each required entry: UsernameToken, Timestamp and Body).

Name	Namespace	Encoding
UsernameToken	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd	Content
Timestamp	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd	Content
Body	http://schemas.xmlsoap.org/soap/envelope/	Content

Table 30: WS-Security Configuration Signature Parts Information



7.8 Invoking Web Service

This section describes how to invoke the Ad-hoc usage query Web Services.

7.8.1 URLs for Invoking Ad-hoc Usage Queries

These URLs should be used for invoking Ad-hoc usage query Web Services:

Staging

https://63.100.30.79:5204/BIM_MeterUsageWeb/sca/WS_MeterUsageExp

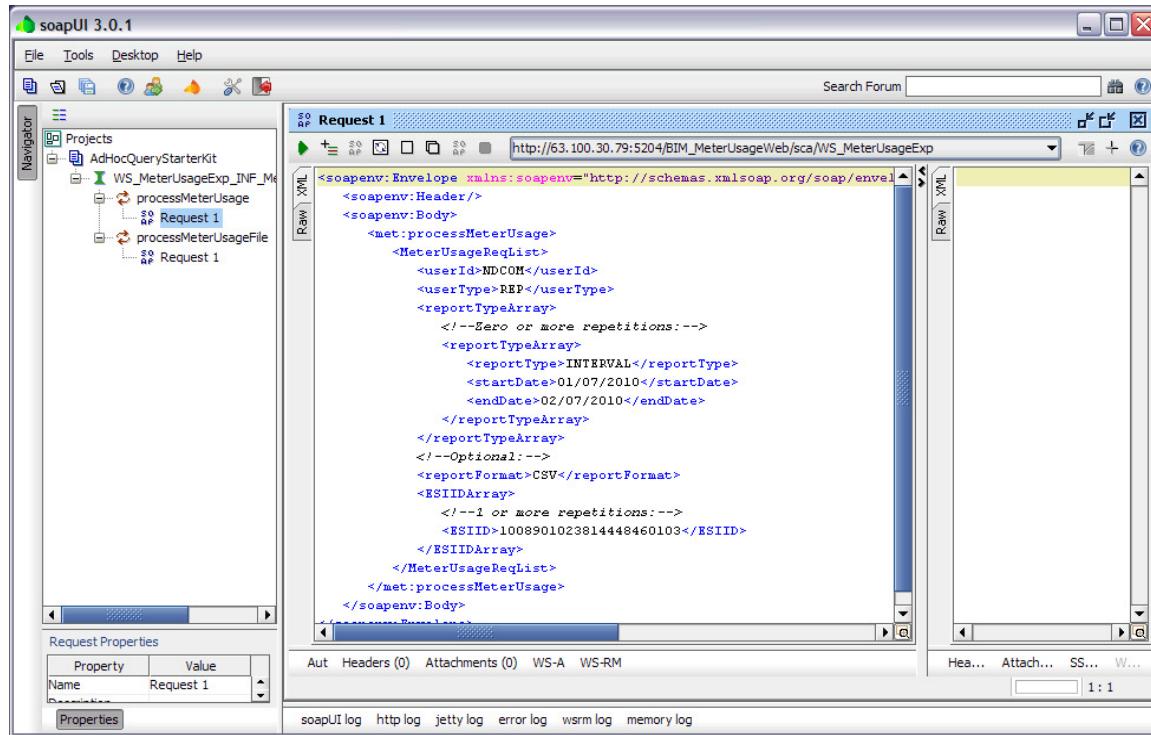
Production

https://services.smartmetertexas.net:5204/BIM_MeterUsageWeb/sca/WS_MeterUsageExp

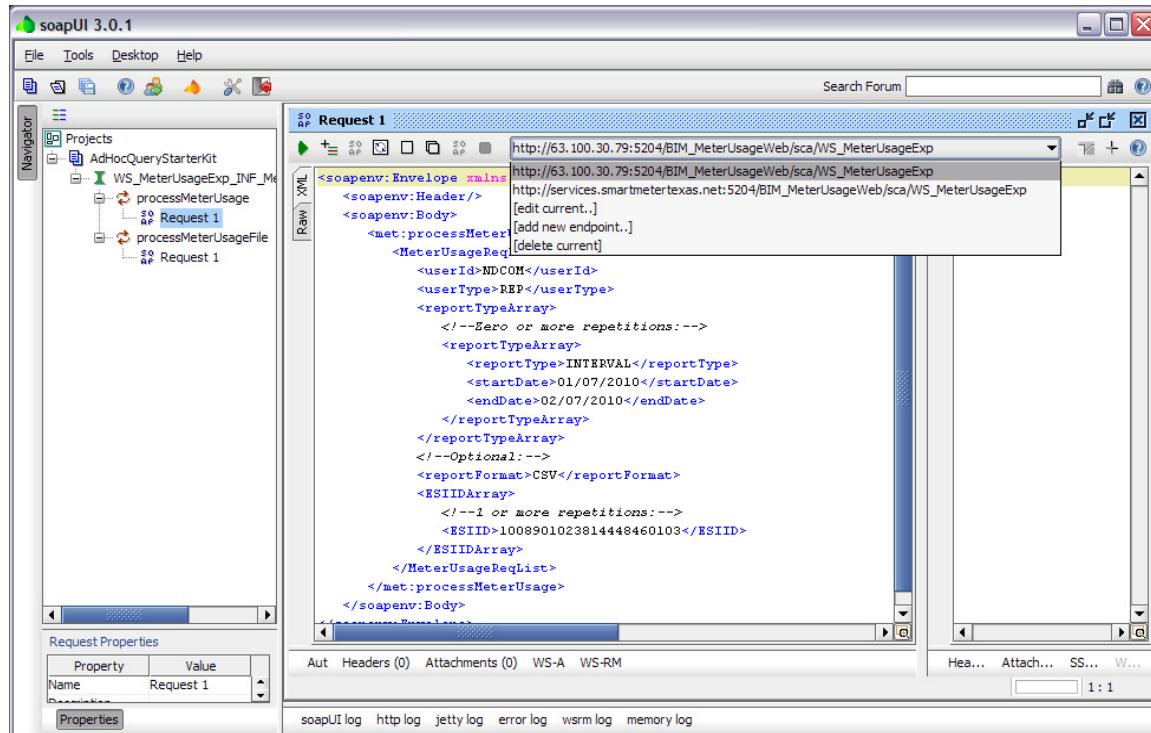
7.8.2 Staging URLs

The appropriate Staging and Production URLs for Ad-Hoc Usage Queries are provided in Section 7.8.1. This section shows how to select the correct URL from within soapUI.

In the figure below, the Staging URL is displayed in the URL or endpoint drop-down box for a Request. This makes it the “current endpoint”. If the request were submitted without modification, it would be sent to the Ad-Hoc Usage Query URL in the SMT staging environment.



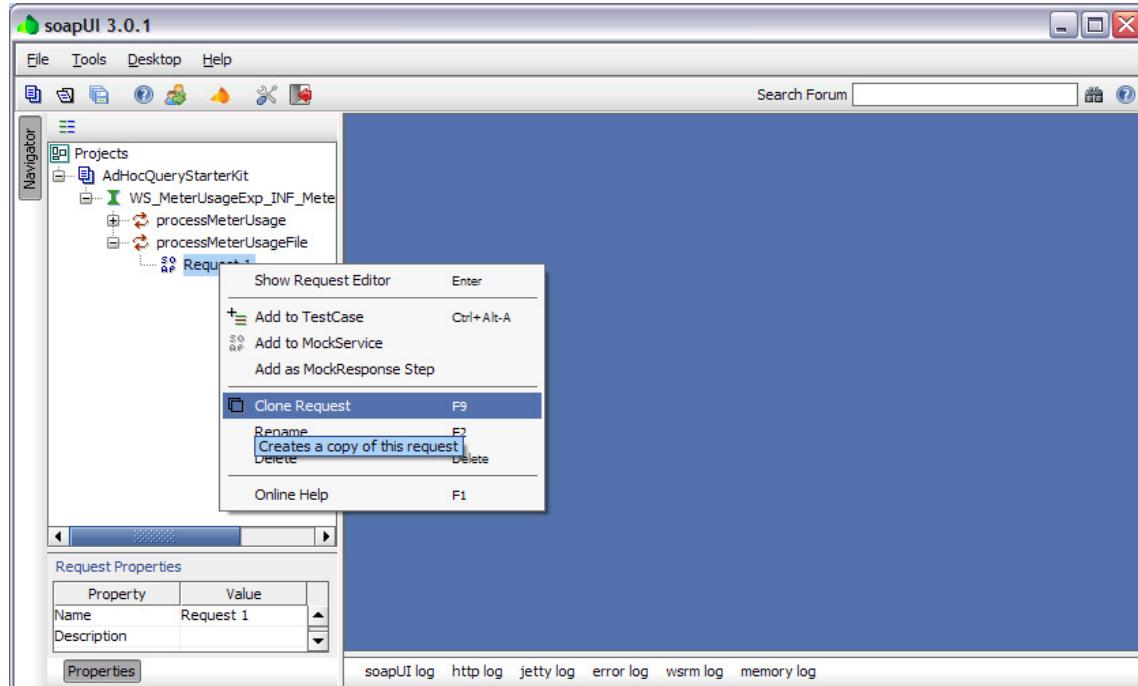
To select the Production end-point, click on the endpoint drop-down. soapUI will present several options. Choose the Production end-point.



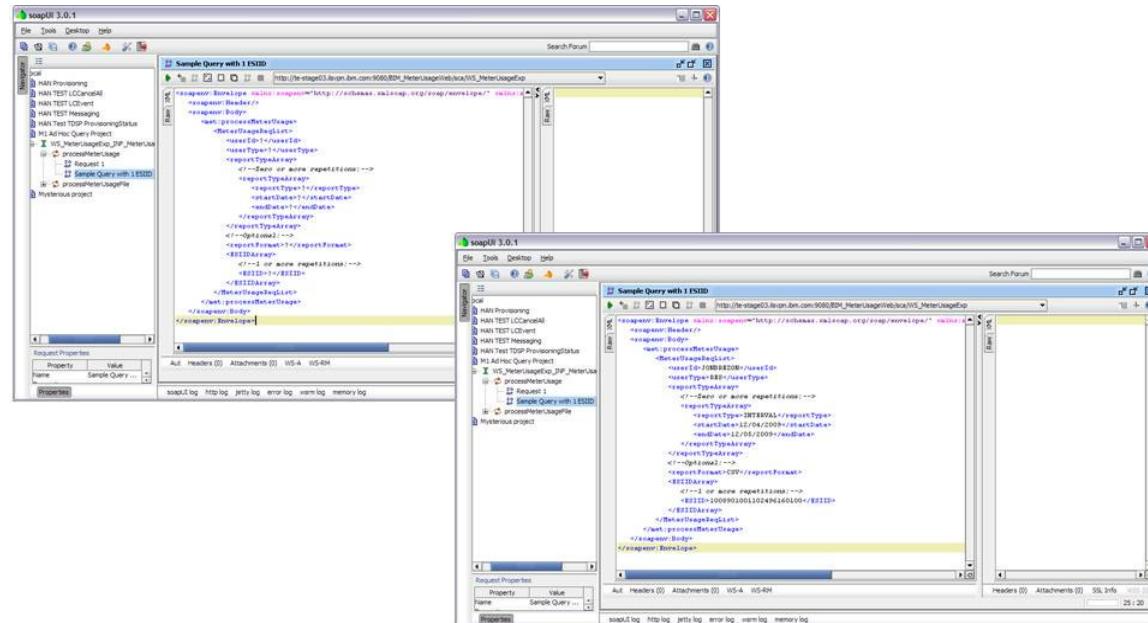
7.8.3 Applying a Web Services Signature to an Outgoing Request

The Starter Kit comes with empty request messages. These empty messages are named *Request 1*. It is recommended that these request messages be kept as starter messages.

To create an Ad-hoc Query request, it is recommended that a new request be created. Right-click on the Request 1 object, select the Clone Request option, and rename the request as depicted below.



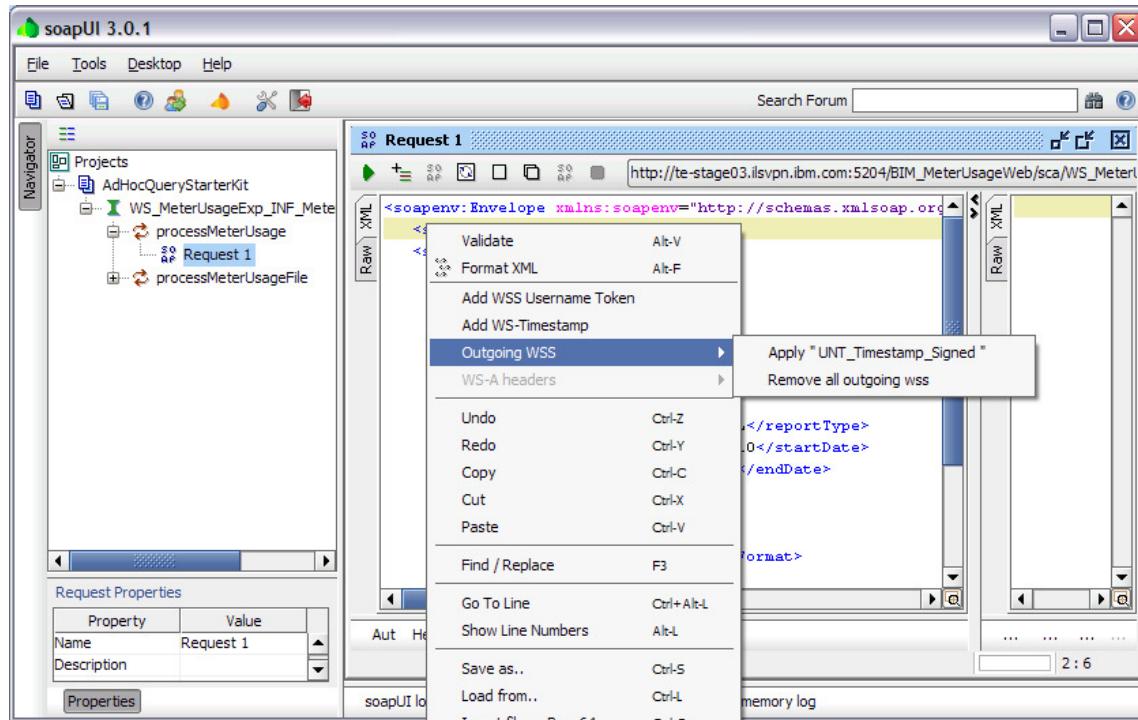
Next, enter the appropriate API data in the request.



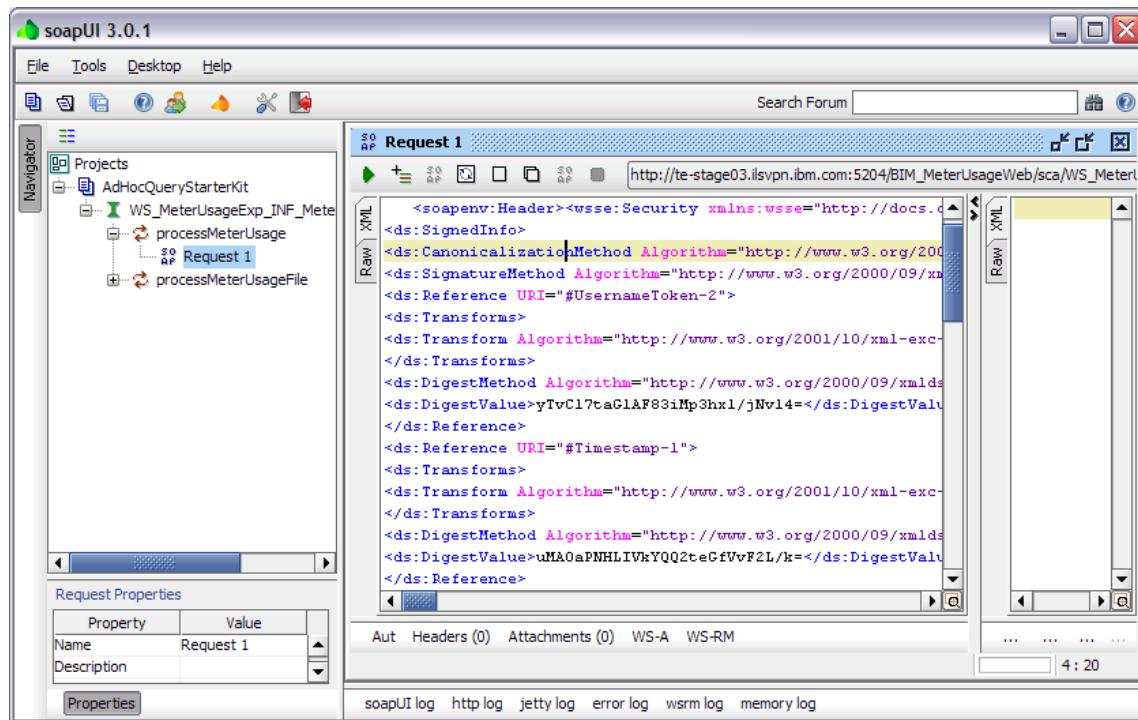
Refer to the API document that is included in the Starter Kit. An example SOAP request appears below.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
    xmlns:met="http://schemas.esb.ams.com/meterusagesource">  
    <soapenv:Header/>  
    <soapenv:Body>  
        <met:processMeterUsage>  
            <MeterUsageReqList>  
                <userId>JONBREZON</userId>  
                <userType>REP</userType>  
                <reportTypeArray>  
                    <!--Zero or more repetitions:-->  
                    <reportTypeArray>  
                        <reportType>INTERVAL</reportType>  
                        <startDate>07/01/2009</startDate>  
                        <endDate>08/09/2009</endDate>  
                    </reportTypeArray>  
                    <reportTypeArray>  
                        <reportType>DAILY</reportType>  
                        <startDate>07/01/2009</startDate>  
                        <endDate>08/09/2009</endDate>  
                    </reportTypeArray>  
                    <!--Optional :-->  
                    <reportFormat>CSV</reportFormat>  
                    <ESIIDArray>  
                        <!--1 or more repetitions:-->  
                        <ESIID>1008901022900060000000</ESIID>  
                    </ESIIDArray>  
                </MeterUsageReqList>  
            </met:processMeterUsage>  
        </soapenv:Body>  
    </soapenv:Envelope>
```

Finally, right click within the message, select Apply Outgoing WSS option, and then select Outgoing WS-Security Configuration.



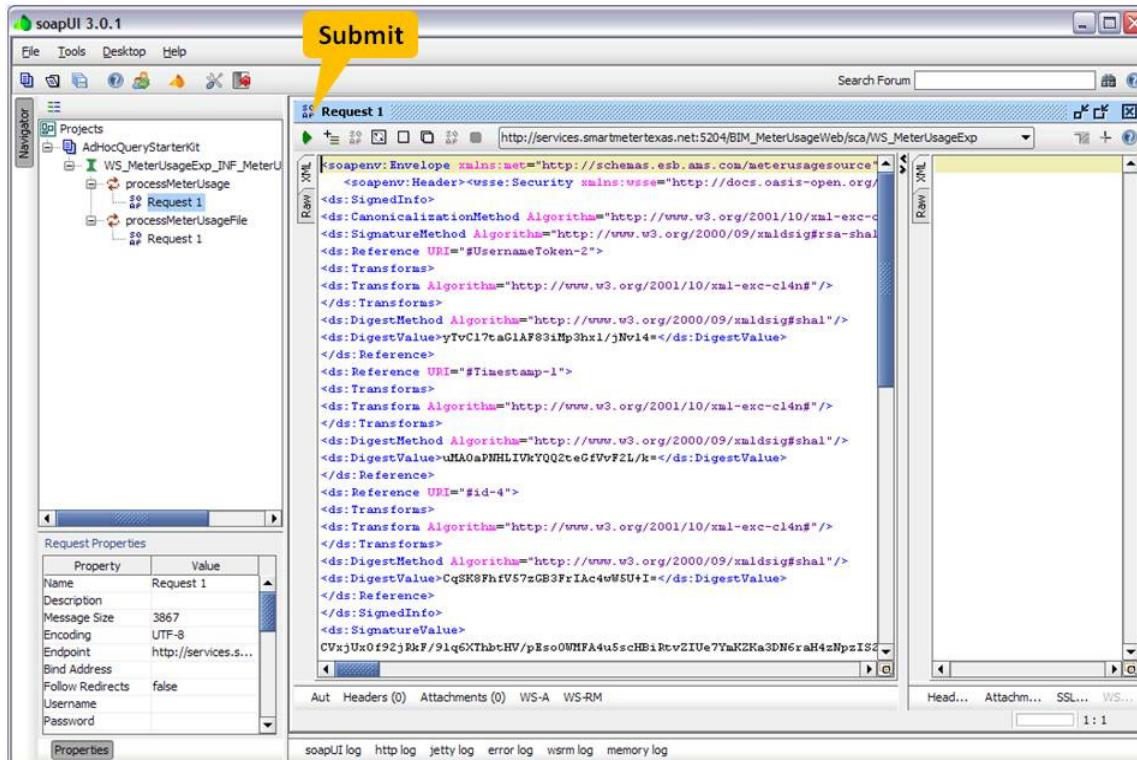
The last step signs the message as pictured below.



Do not reformat the message. If the message is modified, the signature should be removed and reapplied.

Make sure that the correct address is loaded into the URL bar at the top of the request. (See Section 7.8.1 for the URLs for Staging and Production.)

After a Web Services Signature has been applied to the message and the URL is correct, submit message by pressing the Green arrow at the top of the message window.



7.9 Troubleshooting

In the event that Start Kit users are not able invoke a web service via SOAPUI client, these troubleshooting steps should be:

- Review the soapUI error log.

Records in the soapUI error log should indicate a range of communication problems and can be used to determine the root-cause of a failure.

- Check if you can ping the SMT Server, Example: "ping 63.100.30.79" from the machine where the SOAPUI client is running?

If ping succeeds, it means you are able to communicate to the server. Proceed to test telnet as described below.

If you cannot ping the SMT server, soapUI requests are not reaching the SMT server. Determine and fix the cause of the problem that is impeding end-to-end communications.

- Check if you can telnet the SMT Server, Example: "telnet 63.100.30.79 5204" from the machine where the SOAPUI client is running?

If telnet succeeds, it means you are able to communicate to the server using port 5204.

If you cannot ping the SMT server, soapUI requests are not reaching the SMT server. Determine and fix the cause of the problem that is impeding end-to-end communications.

- Check your public SSL key "Common Name" is same as the "hostname" of the server where SOAPUI client is running?

You will need to generate and send a new certificate and submit it to SMT.

- For communications to the Production environment, validate that your public IP address matches with the value originally submitted to the SMT team during REP configuration.

Starter Kit requests will only pass through the SMT firewall when they are submitted from defined IP addresses.

- If the submitted web service request failed with a SOAP Fault Exception, analyze the error code, description in the SOAP response and validate the input message.

SOAP Fault Error codes appear below with a high-level description:

2101=Data cannot be loaded because a mandatory data element is missing

2110=Date Format is invalid. Valid format is MM/DD/YYYY

2111=Start Date should not be greater than currentDate

2112=End Date should not be greater than currentDate

2113=End Date should be greater than startDate

2114=Maximum reportTypeCount should be two

2115=Maximum data limit of duration is 48 months

- If the submitted web service request failed with a SMT Technical error, retry the service again and contact SMT technical support if the problem persists.

SMT Technical Error codes appear below with a high-level description:

2050=Internal error. Could not connect to Database

2080=Non-categorized error. Unexpected Error

The actual SOAP Fault description in the response will contain a detailed message related to the specific input attributes. Check the input elements and retry the service with valid inputs

After you have tried all tests described above but still are not able to submit requests successfully, you may submit requests for help via email to support@smartmetertexas.com. Please collect and send as much diagnostic information as possible with your request, including:

- Information concerning the tests described above
- Information out of the soapUI error log
- The XML request that was submitted and any responses received.

8. Appendices

8.1 Glossary of Terms

Term	Definition
API	Application Programming Interface that allows one program to talk to another
CSV	Comma Separated Values – a common file format in which the data is presented as a list of fields that are separated by commas. CSV files are typically opened with an application or spreadsheet software, such as MS-Excel.
DUNS	Number assigned to a REP by Dunn and Bradstreet. REPS can establish multiple businesses within their parent company using multiple DUNS
ESB	Enterprise Server Bus
EIID	Electric Service Identifier, a unique identifier for the point of delivery
ETL	Extract Transform and Load
FTP	File Transfer Protocol
FTPS	File Transfer Protocol over SSL
HAN	Home Area Network – this is the network between the advanced meter and the home device as mentioned in the AMS rules
HTTP	Hyper Text Transfer Protocol
MQ	Message Queue
REP	Retail Electric Provider
ROR	REP of Record
SMT	Smart Meter Texas
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer is a set of cryptographic protocols that provide security and data integrity for communications over networks such as the Internet
TDSP	Transmission and Distribution Service Provider
WPS	WebSphere Process Server
WSDL	Web Service Description Language - XML-based language for describing Web services
XML	Extensible Markup Language
ZigBee	Specification of a suite of high level communication protocols using small low-power radios

Table 31: Glossary of Terms

8.2 Certification Authorities (CA) List

SMT will accept certificates (SSL and Signer) issued by following Certificate Authorities:

- American Express
- ANX
- Belacom-E-Trust
- C-and-W-HKT-SecureNet-CA
- Certipose
- Certisign
- Certplus
- Deutshe
- Entrust
- Equifax
- EUnet
- FESTE
- First-Data
- GlobalSign
- GTE-CyberTrust
- Microsoft
- NetLock
- RSA
- Saunalahaden
- SecureNet
- SecureSign
- SwissKey
- TC-TrustCenter
- Thawte
- UTN-DATACORP
- Valicert
- Verisign
- ViaCode

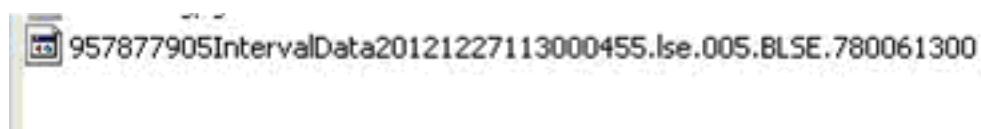
8.3 Decrypting the LSE file

The follow describes the steps to decrypt the LSE files at a local desktop.

Step 1: decrypt the LSE files using your PGP key with a gpg or pgp software



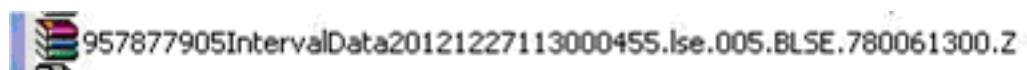
Step 2: when the LSE file is decrypted successfully, you will find the file in compressed format as below:



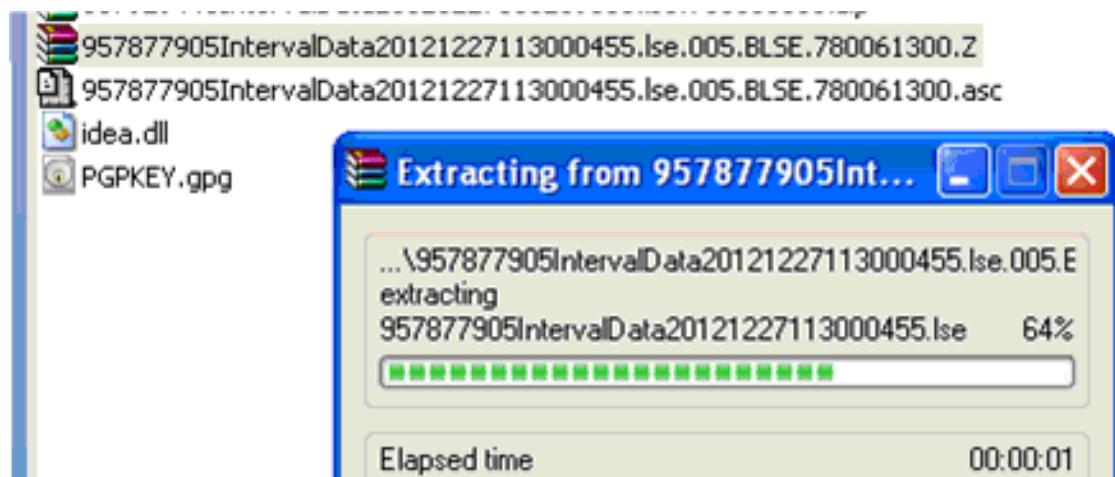
Step 3: rename the file with Z or Zip format

From: 957877905IntervalData20121227113000455.lse.005.BLSE.780061300

To: 957877905IntervalData20121227113000455.lse.005.BLSE.780061300.Z



Step 4: uncompress the file.



Once the LSE file is uncompressed successfully, you will find the original LSE file in a text format which allows you to view the usage data in the file using a text viewing/editing tool such as TextPad.

