****

**CenturyLink EASE ASR Pre-Order Technical Specifications**

Document Release Date: February 2016

Version

Revision History

All versions of this document are listed in chronological order.

|  |  |  |
| --- | --- | --- |
| Version | Date | Description |
| 1.0 | 2/12/16 | Final |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Important: *This document has been through a formal review process. To the best of our knowledge, it is accurate. There may be further modifications and we reserve the right to make those modifications.*

Table of Contents

[Introduction 3](#_Toc433636256)

[Referenced Documents: 3](#_Toc433636257)

[System Requirements 3](#_Toc433636258)

[Definitions 3](#_Toc433636259)

[Supported Services 4](#_Toc433636260)

[Supported ASR Pre-Order Services (Synchronous Services) 4](#_Toc433636261)

[Communication and Security 4](#_Toc433636262)

[ASR Receive Web Service Interface Architecture 5](#_Toc433636263)

[WSDL and ASR Flow 6](#_Toc433636264)

[ASR Request – Receive WSDL 6](#_Toc433636265)

[ASR Response – Send WSDL 7](#_Toc433636266)

[tML Header Requirements 8](#_Toc433636267)

[tML Header Schema 9](#_Toc433636268)

[Text Version of the Schema 9](#_Toc433636269)

[tMLHeader Details Summary 12](#_Toc433636270)

[ASR Pre-Order Receive Positive / Successful 12](#_Toc433636271)

[ASR Pre-Order Receive Negative / Fault 13](#_Toc433636272)

[SOAP Structure 14](#_Toc433636273)

[ASR Pre-Order Synchronous Request 14](#_Toc433636274)

[ASR Pre-Order Synchronous Response 14](#_Toc433636275)

[ASR Pre-Order Synchronous SOAP Fault Response 14](#_Toc433636276)

[Sample ASR tMLs 15](#_Toc433636277)

[ASR Pre-Order Inquiry tML 15](#_Toc433636278)

[ASR Pre-Order Response tML 15](#_Toc433636279)

[SOAP Fault 16](#_Toc433636280)

[SOAP Fault Codes and Fault Messages 17](#_Toc433636281)

# Introduction

The purpose of this document is to provide API specifications to interface with the CenturyLink ASOG Receive Web Service.

## Referenced Documents:

ATIS Documents for UOM ASR

UOM ASR XML Schema

CenturyLink customers must purchase/procure the above-referenced documents from ATIS.

## System Requirements

* Web Services over HTTP/HTTPS
* SOAP version 1.2
* WSDL 1.1

## Definitions

XML – Extensible Markup Language

ASR – Access Service Request

ASOG – Access Services Ordering Guideline

WSDL – Web Service Definition Language

RPC – Remote Procedure Call

HTTP – Hypertext Transfer Protocol

SOAP – Simple Object Access Protocol

tML – Telecommunications Messaging Language

# Supported Services

## Supported ASR Pre-Order Services (Synchronous Services)

UOM ASR XML is used as payload for the ASR Pre-Order request and responses. Following pre-order requests are supported and note that response is returned synchronously.

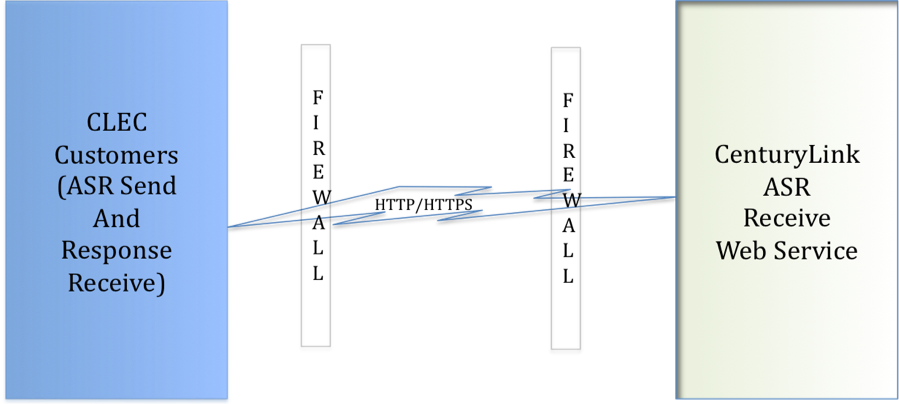
* CFA pre-order Inquiry
* Location pre-order Inquiry
* BAN Inquiry

## Communication and Security

CenturyLink’s gateway to EASE is CXG7-O.  Customer communication with CenturyLink is defined in the CXG7-O questionnaire at the beginning of the project.  This form includes information concerning URLs, IP addresses as well as security.  The questionnaire will be distributed when a Access Customer is being added to UOM.

# ASR Receive Web Service Interface Architecture

ASR Receive Web Service interfaces follow standard business-to-business integration mode between two trading partners.



* CenturyLink ASR Receive Web Service is implemented in RPC style per the WSDL published to CenturyLink customers.
* CenturyLink customers need to implement the Web Service client per the ASR Receive WSDL to send synchronous and asynchronous ASR requests to CenturyLink ASR Receive Web Service.
* CenturyLink customers also need to implement RPC style Web Service per the ASR Response WSDL published by CenturyLink to receive asynchronous ASR order response notifications from CenturyLink ASR Receive service.
* CenturyLink customer ASR send client needs to initiate the SOAP request with tML payload XML (UOM ASR XML) HTTPS and one-way SSL handshaking.
* CenturyLink firewall ensures the request is coming from an authorized source.
* CenturyLink ASR Receive Web Service interface extracts the payload, forwards it to the back end system and sends the SOAP response back to the CenturyLink customer’s service.
  + For orders it is the ACK or NACK
  + For pre-orders it is the pre-order response.

# WSDL and ASR Flow

CenturyLink ASR Receive will support only RPC style Web services at this point. The service is synchronous for ASR pre-order and asynchronous for ASR order. Only one input argument will be taken per service, and it is in xml format. CenturyLink customers must escape the Request payload, i.e., the characters listed below must be replaced by their corresponding equivalent. If the payload data contains any of these special characters, they must be double escaped.

|  |  |
| --- | --- |
| **Character** | **Equivalent** |
| & | &amp; |
| ‘ | &apos; |
| “ | &quot; |
| > | &gt; |
| < | &lt |

The response is in the format of xml string and the payload will be escaped.

## 

## ASR Request – Receive WSDL

CenturyLink ASR Receive Web Service is implemented on the CenturyLink side based on this WSDL. CenturyLink customers need to implement the Web Service client based on this WSDL to send synchronous and asynchronous ASR requests.

<?xml version="1.0" encoding="UTF-8"?>  
<definitions targetNamespace="java:asr.webservice.wisor.com" xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:tns="java:asr.webservice.wisor.com" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
 <message name="request">  
 <part xmlns:partns="http://www.w3.org/2001/XMLSchema" type="partns:string" name="string"/>  
 </message>  
 <message name="response">  
 <part xmlns:partns="http://www.w3.org/2001/XMLSchema" type="partns:string" name="response"/>  
 </message>  
 <portType name="asrport">  
 <operation name="processAsyncRequest">  
 <input message="tns:request"/>  
 <output message="tns:response"/>  
 </operation>  
 <operation name="processSyncRequest">  
 <input message="tns:request"/>  
 <output message="tns:response"/>  
 </operation>  
 </portType>  
 <binding type="tns:asrport" name="asrport">  
 <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>  
 <operation name="processAsyncRequest">  
 <soap:operation soapAction=""/>  
 <input>  
 <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="java:asr.webservice.wisor.com" use="literal"/>  
 </input>  
 <output>  
 <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="java:asr.webservice.wisor.com" use="literal"/>  
 </output>  
 </operation>  
 <operation name="processSyncRequest">  
 <soap:operation soapAction=""/>  
 <input>  
 <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="java:asr.webservice.wisor.com" use="literal"/>  
 </input>  
 <output>  
 <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="java:asr.webservice.wisor.com" use="literal"/>  
 </output>  
 </operation>   
 </binding>  
 <service name="asrservice">  
 <port binding="tns:asrport" name="asrport">  
 <soap:address location="Wisor\_Service\_URL"/>  
 </port>  
 </service>  
</definitions>

## ASR Response – Send WSDL

CenturyLink customers need to implement the Web Service based on this WSDL to receive asynchronous ASR responses (notifications) from the CenturyLink ASR service.

<?xml version="1.0" encoding="UTF-8"?>  
<definitions targetNamespace="java:webservice.asr.customer.wisor.com" xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:tns="java:webservice.asr.customer.wisor.com" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
 <message name="response">  
 <part xmlns:partns="http://www.w3.org/2001/XMLSchema" type="partns:string" name="string"/>  
 </message>  
 <message name="result"/>  
 <portType name="asrport">  
 <operation name="processAsyncResponse">  
 <input message="tns:response"/>  
 </operation>  
 </portType>  
 <binding type="tns:asrport" name="asrport">  
 <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>  
 <operation name="processAsyncResponse">  
 <soap:operation style="rpc" soapAction=""/>  
 <input>  
 <soap:body namespace="java:webservice.asr.customer.wisor.com" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="literal"/>  
 </input>  
 </operation>  
 </binding>  
 <service name="asrservice">  
 <port name="asrport" binding="tns:asrport">  
 <soap:address location="http://accesscustomer.com"/>  
 </port>  
 </service>  
</definitions>

## tML Header Requirements

CenturyLink customers must always send the tML Header on order and pre-order requests. CenturyLink’s ASR Receive Webservice will also send the tML Header on order and pre-order responses back to the customer.

The tML Header will not be sent on positive and negative acknowledgements to the customer in the Request flow and is not expected from the customer on positive and negative acknowledgements in the Response flow.

The ASR Receive Webservice will validate the tML Header on the Request for all elements in the schema including the characteristics mentioned in the table below. If any of tML Header field validations fail, a SOAP fault with a message “Error while validating the tML Header. Please correct the tML Header and retry again” will be returned.

Only the fields mentioned in the table below are supported. The additional fields in the tML Header schema are not used. The corresponding tML Header schema has been provided in this section. Standard responses will only contain the tmlHeader elements ApplicationType, From, and To. Other elements can be returned on an optional basis but are not the standard.

|  |  |  |
| --- | --- | --- |
| **tML Header Field** | **Characteristics** | **Usage** |
| ApplicationType | Required  AN(32) | This field identifies the Application Type.  The predefined ApplicationType value will be configured against the corresponding From-To combination. The standard value will be “ASR” for orders and “ASRPO” for pre-orders.  When a request is received, the web service will validate the ApplicationType on the request with the predefined value configured for that From-To combination and return a SOAP fault if there is no match.  In the response flow, the web service will return the configured Application Type on asynchronous responses.  The web service will not validate the ApplicationType on the acknowledgement (ACK/NACK) from the Access Customer for the asynchronous response. |
| **tML Header Field** | **Characteristics** | **Usage** |
| From | Required  AN(64) | This field identifies the Access Customer. It must be sent by the Access Customer on the request.  The static value to be sent will be predefined by CenturyLink. The standard value will be the Access Customer primary CCNA or ESP name.  On asynchronous responses, this value will be returned under the <To> tag of the tML Header. |
| To | Required  AN(64) | This field identifies CenturyLink on the request flow.  This will be sent by the Access Customer on the request. The static value to be sent will be “CL”.  On asynchronous responses, this value will be returned under the <From> tag of the tML Header. |

## tML Header Schema

### Text Version of the Schema

<?xml version="1.0" encoding="UTF-8"?>  
<!--Generated by Turbo XML 2.3.1.100. Conforms to w3c http://www.w3.org/2001/XMLSchema-->  
<!-- Modified by Madhusudhan K S on 8/26/2014 added ApplicationVersion element and removed sequence in complex element and changed it to all -->  
<xs:schema targetNamespace="http://tml.t1m1.org/tML.Transport.xsd" xmlns="http://tml.t1m1.org/tML.Transport.xsd" xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">  
 <!--Gleaned from TRANSPORT tML.doc, Appendix D -->  
 <!-- HEADER FIELDS -->  
 <xs:element name="TransportID">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:minLength value="1"/>  
 <xs:maxLength value="64"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 <xs:element name="CorrelationID">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:minLength value="1"/>  
 <xs:maxLength value="64"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 <xs:element name="TrackID">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:minLength value="1"/>  
 <xs:maxLength value="64"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 <!-- Dan added this as a guess -->  
 <xs:element name="ResultID">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:minLength value="1"/>  
 <xs:maxLength value="64"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 <!-- end of guess -->  
 <xs:element name="ApplicationType">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:minLength value="1"/>  
 <xs:maxLength value="32"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 <xs:element name="From">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:minLength value="1"/>  
 <xs:maxLength value="64"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 <xs:element name="To">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:minLength value="1"/>  
 <xs:maxLength value="64"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 <xs:element name="SendTimestamp" type="xs:dateTime"/>  
 <xs:element name="RetryCount" type="xs:integer"/>  
 <xs:element name="ApplicationVersion">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:minLength value="1"/>  
 <xs:maxLength value="32"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 <xs:element name="tMLHeader">  
 <xs:complexType>  
 <xs:all>  
 <xs:element ref="TransportID" minOccurs="0"/>  
 <xs:element ref="CorrelationID" minOccurs="0"/>  
 <xs:element ref="TrackID" minOccurs="0"/>  
 <xs:element ref="ApplicationType" minOccurs="0"/>  
 <xs:element ref="From"/>  
 <xs:element ref="To"/>  
 <xs:element ref="SendTimestamp" minOccurs="0"/>  
 <xs:element ref="RetryCount" minOccurs="0"/>  
 <xs:element ref="ApplicationVersion" minOccurs="0"/>  
 </xs:all>  
 </xs:complexType>  
 </xs:element>  
 <!-- OK MESSAGE -->  
 <xs:element name="OK">  
 <xs:complexType/>  
 </xs:element>  
 <!-- REQUEST ACKNOWLEDGEMENT FIELDS -->  
 <xs:element name="RequestAcknowledgement">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="Acknowledgement" type="xs:boolean"/>  
 <xs:element name="Reason" type="xs:string" minOccurs="0"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <!-- REQUEST NOTIFICATION FIELDS -->  
 <xs:element name="Result">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element ref="ResultID"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <xs:element name="ResultWaitingNotification">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element ref="Result" minOccurs="0" maxOccurs="unbounded"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <!-- GET RESULT FIELDS -->  
 <xs:element name="GetResult">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element ref="ResultID"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <!-- RESULT ACKNOWLEDGEMENT FIELDS -->  
 <xs:element name="ResultAcknowledgement">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element ref="ResultID"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <!-- EXCEPTION DESCRIPTION FIELDS -->  
 <xs:element name="ExceptionDescription" type="ExceptionDescription\_Type"/>  
 <xs:complexType name="ExceptionDescription\_Type">  
 <xs:sequence>  
 <xs:element name="Type" type="ExceptionType\_Type"/>  
 <xs:element ref="ExceptionDetails" minOccurs="0" maxOccurs="unbounded"/>  
 </xs:sequence>  
 </xs:complexType>  
 <xs:element name="ExceptionDetails" type="ExceptionDetails"/>  
 <xs:complexType name="ExceptionDetails">  
 <xs:sequence>  
 <xs:element name="Code" minOccurs="0">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:minLength value="1"/>  
 <xs:maxLength value="64"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 <xs:element name="Description">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:minLength value="1"/>  
 <xs:maxLength value="64"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 <xs:element name="AdditionalInfo" minOccurs="0">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:minLength value="1"/>  
 <xs:maxLength value="240"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 <xs:simpleType name="ExceptionType\_Type">  
 <xs:restriction base="xs:string">  
 <xs:enumeration value="GENERAL\_FAILURE"/>  
 <xs:enumeration value="GENERAL\_ERROR\_RETRYABLE"/>  
 <xs:enumeration value="TRANSPORT\_FAILURE"/>  
 <xs:enumeration value="TRANSPORT\_ERROR\_RETRYABLE"/>  
 <xs:enumeration value="SECURITY\_PROBLEM"/>  
 <xs:enumeration value="RESOURCE\_PROBLEM\_RETRYABLE"/>  
 <xs:enumeration value="FUNCTION\_NOT\_SUPPORTED"/>  
 <xs:enumeration value="INVALID\_MESSAGE\_FORMAT"/>  
 <xs:enumeration value="RETURN\_RESULT\_PROBLEM"/>  
 <xs:enumeration value="DUPLICATE\_INVOCATION"/>  
 <xs:enumeration value="INVALID\_ATTRIBUTE"/>  
 <xs:enumeration value="MISSING\_ATTRIBUTE"/>  
 <xs:enumeration value="INVALID\_DATA\_VALUE"/>  
 </xs:restriction>  
 </xs:simpleType>  
</xs:schema>

### tMLHeader Details Summary

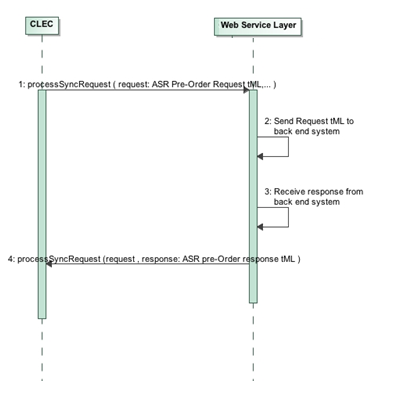
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Max** | **Required** | **Value** |
| ApplicationType | String | 32 | Y | “ASR” or “ASRPO” |
| From | String | 64 | Y | Primary CCNA or ESP name |
| To | String | 64 | Y | “CL” |

## ASR Pre-Order Receive Positive / Successful

CenturyLink ASR Send customers (referred to an Access Customer here) must send a SOAP request with ASR UOM Pre-Order XML as a string *request* parameter over the operation *processSyncRequest* with a place holder for the *response* string parameter.

The Pre-Order Request is sent to the back end system for further processing while still holding on to the web service connection with the Access Customer.

When response data is received from the back end system, the Web Service Layer notifies the response to the Access Customer as a tML in the *response* message of the same *processSyncRequest* web service call.

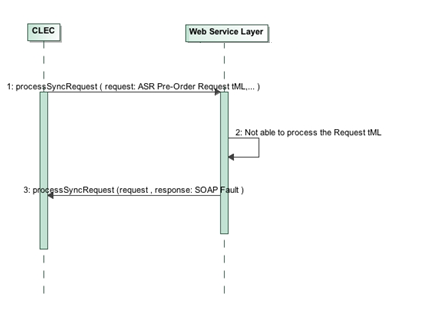


## ASR Pre-Order Receive Negative / Fault

CenturyLink ASR Send customers (referred to as Access Customer here) must send a SOAP request with ASR UOM Pre-Order XML as a string *request* parameter over the operation *processSyncRequest* with a place holder for the *response* string parameter.

CenturyLink system is not able to handle the Pre-Order Request.

A SOAP Fault is sent to the Access Customer in the *response* message of the same *processSyncRequest* web service call.



# SOAP Structure

## ASR Pre-Order Synchronous Request

<?xml version="1.0" encoding="UTF-8"?>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<soapenv:Header>

<tMLHeader xmlns="http://tml.t1m1.org/tML.Transport.xsd">

<ApplicationType>ASR</ApplicationType>

<From> Access CustomerESP</From>

<To>CL</To>

</tMLHeader>

</soapenv:Header>

<soapenv:Body>

<processSyncRequest xmlns="java:asr.webservice.wisor.com">

<string xmlns="">XML GOES HERE!!!</string>

</processSyncRequest>

</soapenv:Body>

</soapenv:Envelope>

## ASR Pre-Order Synchronous Response

<?xml version="1.0" encoding="utf-8"?>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<soapenv:Header>

<tMLHeader xmlns="http://tml.t1m1.org/tML.Transport.xsd">

<ApplicationType>ASR</ApplicationType>

<From>CL</From>

<To> Access CustomerESP</To>

</tMLHeader>

</soapenv:Header>

<soapenv:Body>

<processSyncRequestResponse xmlns="java:webservice.asr.customer.wisor.com">

<response xmlns=""> XML GOES HERE!!!</response>

</processSyncRequestResponse>

</soapenv:Body>

</soapenv:Envelope>

## ASR Pre-Order Synchronous SOAP Fault Response

<?xml version="1.0" encoding="utf-8"?>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<soapenv:Body>

<soapenv:Fault>

<faultcode>WG1602</faultcode>

<faultstring>There is no interface configured, contact Wisor support</faultstring>

<detail>

<ns1:hostname xmlns:ns1="http://xml.apache.org/axis/">in-bandevs04</ns1:hostname>

</detail>

</soapenv:Fault>

</soapenv:Body>

</soapenv:Envelope>

# Sample ASR tMLs

## ASR Pre-Order Inquiry tML

<?xml version="1.0" encoding="UTF-8"?>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<soapenv:Header>

<ns1:tMLHeader xmlns:ns1="http://tml.t1m1.org/tML.Transport.xsd">

<ns1:ApplicationType>ASR</ns1:ApplicationType>

<ns1:From>ACCESSCUSTOMERESP</ns1:From>

<ns1:To>CL</ns1:To>

</ns1:tMLHeader>

</soapenv:Header>

<soapenv:Body>

<processSyncRequest xmlns="java: asr.webservice.wisor.com">

<string xmlns="">&lt;?xml version=&quot;1.0&quot;?&gt;

&lt;CFA\_INQUIRY xmlns=&quot;http://www.atis.org/OBF/ASR/UOM-ASRInquiry&quot;&gt;

&lt;HDR&gt;

&lt;MESSAGE\_ID&gt;ESB-03&lt;/MESSAGE\_ID&gt;

&lt;CCNA&gt;ALN&lt;/CCNA&gt;

&lt;MSG\_TIMESTAMP&gt;2010-02-11T08:15:18-08:00&lt;/MSG\_TIMESTAMP&gt;

&lt;ICSC&gt;LM10&lt;/ICSC&gt;

&lt;/HDR&gt;

&lt;CFA&gt;

&lt;FACDESG&gt;101&lt;/FACDESG&gt;

&lt;FACTYPE&gt;ST01&lt;/FACTYPE&gt;

&lt;LOCA&gt;LTRTWVXAH00&lt;/LOCA&gt;

&lt;LOCZ&gt;NWHNWVXAH00&lt;/LOCZ&gt;

&lt;/CFA&gt;

&lt;/CFA\_INQUIRY&gt;

</string>

</processSyncRequest>

</soapenv:Body>

</soapenv:Envelope>

## ASR Pre-Order Response tML

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<soapenv:Header>

<ns1:tMLHeader xmlns:ns1="http://tml.t1m1.org/tML.Transport.xsd">

<ns1:ApplicationType>ASR</ns1:ApplicationType>

<ns1:From>CL</ns1:From>

<ns1:To>ACCESSCUSTOMERESP</ns1:To>

</ns1:tMLHeader>

</soapenv:Header>

<soapenv:Body>

<processSyncRequestResponse xmlns="java: webservice.asr.customer.wisor.com">

<response xmlns="">

&lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;

&lt;CFA\_RESULTS xmlns=&quot;http://www.atis.org/OBF/ASR/UOM-ASRInquiry&quot; xmlns:tML=&quot;urn:int.itu/tML/tMLSchemaMetadata&quot;&gt;

&lt;HDR&gt;

&lt;MESSAGE\_ID&gt;14&lt;/MESSAGE\_ID&gt;

&lt;CCNA&gt;ALN&lt;/CCNA&gt;

&lt;ICSC&gt;BSAS&lt;/ICSC&gt;

&lt;MSG\_TIMESTAMP&gt;2008-06-17T11:19:01-04:00&lt;/MSG\_TIMESTAMP&gt;

&lt;IRI&gt;A&lt;/IRI&gt;

&lt;IRM&gt;irm&lt;/IRM&gt;

&lt;/HDR&gt;

&lt;CFA\_DETAILS&gt;

&lt;CFA&gt;

&lt;FACDESG&gt;FDFDF&lt;/FACDESG&gt;

&lt;FACTYPE&gt;DFDFDF&lt;/FACTYPE&gt;

&lt;CHANNEL&gt;FDDF&lt;/CHANNEL&gt;

&lt;LOCA&gt;DFDFD&lt;/LOCA&gt;

&lt;LOCZ&gt;DFDFDF&lt;/LOCZ&gt;

&lt;/CFA&gt;

&lt;CHANNEL\_INFO&gt;

&lt;STATUS&gt;status&lt;/STATUS&gt;

&lt;ECCKT&gt;ecckt&lt;/ECCKT&gt;

&lt;CKR&gt;ckr&lt;/CKR&gt;

&lt;PA&gt;pa&lt;/PA&gt;

&lt;DD&gt;dd&lt;/DD&gt;

&lt;PON&gt;pon&lt;/PON&gt;

&lt;/CHANNEL\_INFO&gt;

&lt;/CFA\_DETAILS&gt;

&lt;/CFA\_RESULTS&gt;</response>

</processSyncRequestResponse>

</soapenv:Body>

</soapenv:Envelope>

## SOAP Fault

<?xml version="1.0" encoding="UTF-8"?>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<soapenv:Body>

<soapenv:Fault>

<faultcode>WG1602</faultcode>

<faultstring>There is no interface configured, contact Wisor support</faultstring>

<detail>

<ns1:hostname xmlns:ns1="http://xml.apache.org/axis/">in-bandevs04</ns1:hostname>

</detail>

</soapenv:Fault>

</soapenv:Body>

</soapenv:Envelope>

# SOAP Fault Codes and Fault Messages

The CenturyLink SOAP fault codes and fault messages are listed below.

|  |  |
| --- | --- |
| **Fault Codes** | **Fault Messages** |
| WG0204 | Empty message received from internal system |
| WG0503 | Gateway is busy. Please retry after sometime. |
| WG1601 | There is no interface configured for Pre-Order. Please contact support. |
| WG1602 | There is no interface configured. Please contact Synchronoss support |
| WG1604 | Interface has been Suspended. Please contact Synchronoss support. |
| WG0641 | Host is not reachable. Please check the URL Configured in the Interface. |
| WG0642 | Server busy, retry request after some time. (This error occurs when backend OSS is not responding in case of pre-order) |
| WG1697 | Translator is deactivated. Please contact Synchronoss support. |
| WSASRTML001 | Gateway not reachable. Please contact Synchronoss support. |
| WSASRTML002 | Internal server error. Please contact Synchronoss support. |
| WSASRTML003 | Server busy, retry request after some time. |
| WSASRTML004 | Unknown error, Please contact Synchronoss support  (This will be sent when EP returns null or blank instead of WG0200) |
| WSASRTML005 | Schema validation failed (the actual validation error will go here) |
| WSASRTML006 | Error while validating the tML Header. Please correct the tML Header and retry again. |
| WSASRTML007 | Error while Transformation. |
| WSASRTML008 | Customer with given From/To is not configured. |
| WSASRTML009 | Username/Password in Security Header is invalid. |
| WSASRTML010 | No Username/Password configured for the Access Customer, Please contact Support. |
| WSASRTML011 | CCNA-ICSC combination is not valid. |