Oracle® Communications MetaSolv Solution ASR

Web Services Developer's Guide

Release 70.0
G19391-01
January 2025

Oracle Communications MetaSolv Solution ASR Web Services Developer's Guide, Release 70.0

G19391-01

Contents

2 ASR Web Service Reference

Preface

This guide describes the Oracle Communications MetaSolv Solution Access Service Request (ASR) Web Services. These web services are provided with ASR 55 or later and are compatible with MetaSolv Solution 6.3.x.

Audience

This guide is intended for developers who have a working knowledge of web services in general, and who understand XML and Java development, including standard Java practices and J2EE principles. This document is for developers who are integrating MetaSolv Solution ASR with Oracle products, or with other external systems. This guide is intended for integration for users who are responsible for submitting ASRs (based on industry guidelines) to a service provider to initiate or change access services.

Before designing any software that interfaces with the ASR Web Services, you should familiarize yourself with the contents of the Access Service Request Guidelines, which are available at:

http://www.atis.org

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

Web Services Overview

This chapter provides introductory information about the Oracle Communications MetaSolv Solution (MSS) Access Service Request (ASR) Web Services.

Overview

Web services support interoperable system-to-system interaction over a network. They are APIs that can be accessed over a network and run on a remote system hosting the requested services. Web services are described by the Web Service Definition Language (WSDL).

This document describes how to integrate MetaSolv Solution ASR with other Oracle products or with external applications using web services. In this book, you can find deployment information and a reference chapter for each web service.

About ASR Web Services

ASR Web Services include two web services that address various ASR features such as importing and exporting different request types of ASR orders. These operations cover the operations that are offered by the CORBA APIs.

ASR Web Service

You can use the ASR Web Service operations to request the following:

- Import and export various request types of ASR orders
- Import and export a Confirmation Notice
- Import and export a Clarification Request
- Import and export a Provider Test Acceptance
- Query ASRs by criteria
- Import an External DLR

You can import or export the following types of ASR orders:

- Transport
- Trunking
- SES
- FGA
- PIP
- DIS
- EUSA
- Ring
- WAL

Refer to "ASR Web Service Reference" for information about the ASR Web Service set of operations.

CABS Web Service

You use the CABS Web Service to export and query billing information for a given ASR. For example, you can request the following:

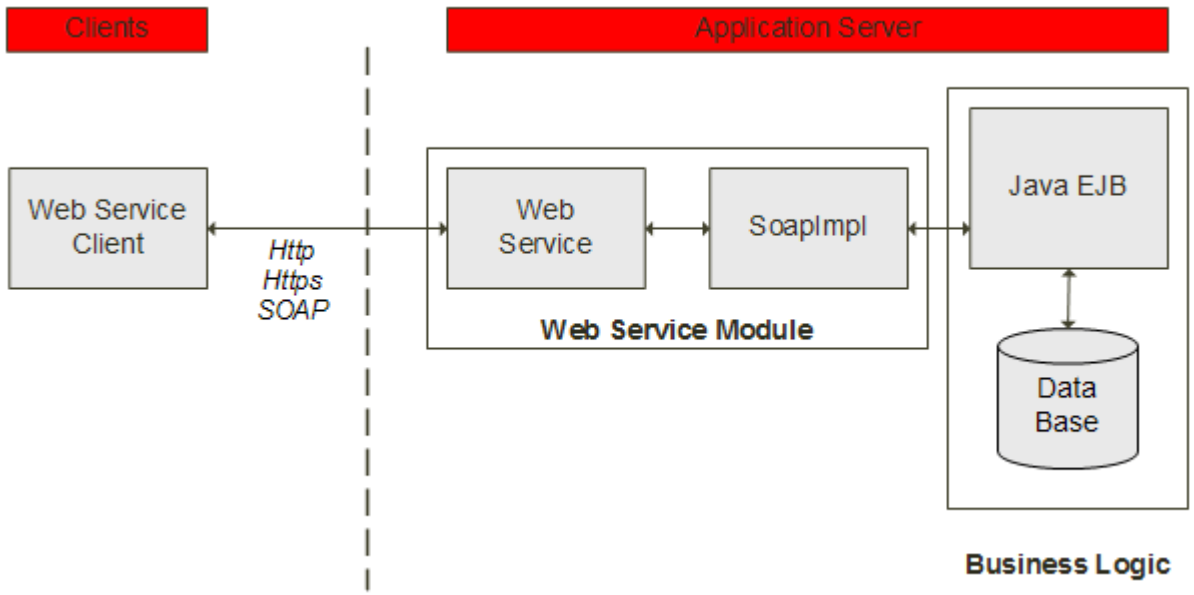
- Export of CABS
- Query CABS by various criteria

Refer to "CABS Web Service Reference" for information about the CABS Web Service set of operations.

About the ASR Web Services Framework

Figure 1-1 shows the path traveled by a call originating from the web service client.

Figure 1-1 ASR Web Services Framework



The path of the web service includes:

- Web service client

  This client represents the web service user. Web service operations are called by sending SOAP messages over HTTP or HTTPS.

- Web service module

  This module represents all the sub-modules required for implementing a web service, for instance, the web service framework, the WSDL interfaces, and the WSDL implementations. The web service module is deployed as a WAR file.

  See "About the ASR Web Service Module" for more information.

- Business logic

  The business logic includes all the sub-modules required for implementing business functionality and includes the Java EJBs and the database.

About the ASR Web Service Module

The web service module is installed as part of the ASR installation. All web services are included in the following EAR files:

- **ASR***release***.ear** file for a single server environment
- **cluster-ASR***release***.ear** file for a clustered server environment

where *release* is the ASR release number. When the ASR installer copies the **EAR** file to the deploy directory, the following modules of the ASR Web Services are deployed and ready to use:

- ASR Web Service
- CABS Web Service

You can deploy two different ASR versions at the same time.

About ASR Web Service Standards and Specifications

See *MSS Web Services Developer's Guide* for information on the web service standards and specifications topic.

About Transaction Handling

Transactions are handled in the same way that MSS Web Services manages transactions. See *MSS Web Services Developer's Guide* for information on this topic.

About Exception Stacktraces

Exception stacktraces are available in the WebLogic Server logs. The error messages are thrown by the operation when an exception occurs. This information is available in the following log files:

- *serverName*.**mss.log** file where *serverName* is the server name of the ASR application installation.
- **AppServerLog_ASR***release***.xml** (for MSS version 6.3.1.452 or earlier) or **AppServer_ASR***release***.log** (for MSS version 6.3.1.558 or later) file where *release* is the ASR release number.

These files are located in the MSS **log** directory.

ASR throws exceptions of type WSException_Exception from the web service. You must handle these exceptions from the client-side code.

Understanding ASR Web Services

Web services are defined by WAR, WSDL, and schema files. For web services, there is a WAR file. The WAR file includes a single WSDL file and numerous schema files. The WSDL files contain the actual web service definitions. The schema files include definitions of specific elements, complex types, and simple types.

About WAR and WSDL Files

The WAR files are in the root directory of the ASR EAR file. Within each WAR file, the WSDL file is located in the **WEB-INF/wsdls** directory. The schemas are also available in the **asr_webservice_schemas.jar** file. You find this JAR file as a deliverable with the ASR installer. Table 1-1 provides the WSDL and WAR file names for the web services.

Table 1-1 WAR and WSDL Files for Each Web Service

Web Service	WAR File Name	WSDL File Name
ASR Web Service	asrws.war	ASRWebService.wsdl
CABS Web Service	cabsws.war	CABSWebService.wsdl

About WSDL Definitions

The WSDL definitions at the beginning of the WSDL file define the web service as being deployed using the SOAP 1.1 protocol over HTTP.

See the following websites for more information about WSDL 1.1 and SOAP 1.1:

- https://www.w3.org/TR/wsdl
- https://www.w3.org/TR/2000/NOTE-SOAP-20000508/

See "About ASR Web Service Standards and Specifications" for more information about the standards that apply to the ASR Web Service.

Example 1-1 shows the definitions section of the **ASRWebService.wsdl** file.

Example 1-1 WSDL Definitions Example

```
<definitions targetNamespace="http://www.openuri.org/" name="ASRWebService"
             xmlns="http://schemas.xmlsoap.org/wsdl/"
             xmlns:tns="http://www.openuri.org/"
             xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
             xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata">
```

About WSDL Locations

For the ASR Web Services, you can access the WSDL information with URL locations. Table 1-2 provides the URL locations for each web service.

Table 1-2 WSDL URL Location for each Web Service

Web Service	URL Location
ASR Web Service	`http://asrHost:port/ASRreleaseWS/asr/ASRreleaseWebService?WSDL`
CABS Web Service	`http://asrHost:port/CabsreleaseWS/asr/CABSreleaseWebService?WSDL`

where:

- *asrHost* is the name of the host on which ASR is installed.

- *port* is the port number of the system on which ASR is installed.
- *release* is the ASR release number because two different versions of ASR can be deployed simultaneously.

About WSDL Bindings

In the WSDL file, the bindings define the protocol details and message formats for the web service operations. Example 1-2 shows a portion of the WSDL file with the binding definition of the exportASR operation.

Example 1-2 ASRSoap Binding Definition Example with exportASR

```
<binding name="ASRWSSoapBinding" type="tns:ASRImpl">
   <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
      style="document"/>
   <operation name="exportASR">
   <soap:operation soapAction="http://www.openuri.org/exportASR"
      style="document"/>
   <input>
      <soap:body use="literal"/>
   </input>
   <output>
       <soap:body use="literal"/>
   </output>
       <fault name="WSException">
           <soap:fault name="WSException" use="literal"/>
       </fault>
   </operation>
</binding>
```

About Namespaces

Each WSDL file defines a namespace to avoid naming conflicts. The namespace definition appears after the WSDL definitions section.

You use the namespace to determine the schema file location of the schema reference. Example 1-3 shows how a namespace defined in the WSDL correlates to the supporting schema files.

In this example, the ASR Web Service WSDL file named **ASRWebService.wsdl** defines and references the **asr** namespace. This excerpt shows exportASRReq has a namespace designation and how the namespace is defined:

Example 1-3 ASRWebService.wsdl Namespace Example

```
.
.
.
<xsd:schema xmlns:asr="http://xmlns.oracle.com/communications/asr">
     <!-- ============= Schema Imports ==================== -->
     <xsd:import namespace="http://xmlns.oracle.com/communications/asr"
                 schemaLocation="../schemas/ASRWebService_schema.xsd"/>
</xsd:schema>
.
.
.
<xsd:element name="exportASRReq">
     <xsd:complexType>
          <xsd:sequence>
              <xsd:element ref="asr:exportASR"/>
```

```
            </xsd:sequence>
        </xsd:complexType>
</xsd:element>
```

The exportASRReq element declaration tells you that exportASR is defined in the schema file that supports the specified namespace. A search for the specified namespace reveals that the **asr** namespace represents the **ASRWebService_schema.xsd** schema file.

After you determine that the **ASRWebService_schema.xsd** schema file defines the XML structure that the WSDL file references, you can navigate through the schema files to determine child XML structures.

Refer to the following website for more information on namespaces:

https://www.w3.org/TR/REC-xml-names/

Understanding ASR WSDL Operations

The WSDL file defines the web service and its operations. It also defines the input, output and fault messages of each operation. ASR WSDL operations follow the request-response or round-trip pattern. When the client sends a request message to the web service, the operation either sends a response message back or the operation sends a fault message back to the client for an error.

In the WSDL file, each service is bound to a port. For example, in the **ASRWebService.wsdl** file, the ASR Web Service is bound to the ASR*release*WSSoap port, where *release* is the ASR release number. Example 1-4 shows the definition of the service and the port name.

Example 1-4 Service and Port Definition

```
<service name="ASRreleaseWebService">
    <port name="ASRreleaseWSSoap" binding="tns:ASRWSSoapBinding">
        <soap:address location="http://localhost:7001/ASRreleaseWS/ASR"/>
    </port>
</service>
```

In Example 1-4, the *release* placeholder is the ASR release number.

The service and port declaration reference the portType through the soap definition. The portType element combines multiple message elements to define the list of web service operations and their parameters.

Example 1-5 demonstrates the following aspects of the portType definition in the ASR Web Service:

- The portType element definition of the name as ASR*release*WSSoap where *release* is the ASR release number.
- The definition of exportASRReq as a round-trip operation
- The declaration of the input, output, and fault message types for exportASRReq

Example 1-5 The WSDL portType and exportASRReq Definition

```
<portType name="ASRImpl">
    <operation name="exportASR">
        <input message="tns:exportASR"/>
        <output message="tns:exportASRResponse"/>
        <fault message="tns:WSException" name="WSException"/>
    </operation>
.
.
```

```
.
</portType>
```

Message definitions determine the input and output parameters for the operation. Example 1-6 shows a WSDL excerpt with the message definitions and their respective elements.

Example 1-6 Web Service Message Definitions for exportASRReq Operation

```
<message name="exportASR">
    <part name="parameters" element="tns:exportASRReq"/>
</message>
<message name="exportASRResponse">
    <part name="parameters" element="tns:exportASRReqResponse"/>
</message>
<message name="WSException">
    <part name="fault" element="tns:WSExceptionRes"/>
</message>
```

Table 1-3 summarizes the definitions in Example 1-5 and Example 1-6 to list the message types, messages, and elements for the exportASRReq operation.

Table 1-3 exportASRReq Messages and Elements

Message Type	Message	Element
input	exportASR	exportASRReq
output	exportASRResponse	exportASRResponse
fault	WSException	WSExceptionRes

The details of the element for each message are defined in declarations in the WSDL file. For example, the input request, output response, and exception/fault elements listed in Table 1-3 are defined by the declarations shown in Example 1-7.

Example 1-7 Element Definitions for Input, Output, and Fault for exportASRReq

```
<xsd:element name="exportASRReq">
     <xsd:complexType>
          <xsd:sequence>
               <xsd:element ref="asr:exportASR"/>
          </xsd:sequence>
     </xsd:complexType>
</xsd:element>
<xsd:element name="exportASRReqResponse">
     <xsd:complexType>
         <xsd:sequence>
             <xsd:element ref="asr:exportASRResponse"/>
         </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="WSExceptionRes">
    <xsd:complexType>
         <xsd:sequence>
             <xsd:element ref="asr:WSException"/>
         </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

The elements include references to the locations in schema files where they are fully defined. The references include namespaces that you use to locate the schema file for each element.

For example, the namespace **asr** was defined earlier to point to the **ASRWebService_schema.xsd** file.

The schema files can themselves include references to other schema files.

Refer to "About Namespaces" for more information on namespaces, and "About Schema Files" for more information on schema files.

About Schema Files

Numerous schema files support the ASR Web Services. Within a WAR file, you find the schema files located in the **WEB-INF/wsdls** directory. The schemas are also available in the **asr_webservice_schemas.jar** file. This JAR file is provided along with the ASR installer file. These schemas are categorized as ASR schemas, CABS schemas, External DLR schemas, and Query ASR schemas.

Refer to the individual web service chapter for more information about schema files.

Getting Information about ASR WSDL Operations

This guide contains high-level documentation about each web service and its operations.

There is a reference chapter dedicated to each web service. Each web service chapter includes an overview of the web service and a section for each operation. Each operation section contains the following:

- An overview of the operation's purpose
- High-level schema information for the input or request message payload
- High-level schema information for the output or response message payload

You use the XSD schema files as reference for field-level information such as descriptions and field formatting guidance. Also, refer to "Using the Access Service Ordering Guidelines" for additional field-level documentation.

Using the Message Payload Documentation

The reference chapter for the web service includes sections for the request and response documentation. For each operation's request and response, a table describes the primary XSD elements and types. An operation can have multiple tables describing complex payloads.

These tables enable you to navigate the request or response in the following ways:

- They provide an overall view of the payload.
- They aid in locating the request or response definitions and their XSD file location.
- They show you XSD definitions, for instance, whether they are elements or complex types.

The table describing a request contains the following information:

- The input request element defined in the first row.
- A row for each element, describing how the element or complexType is defined in the XSD, its type, and the file name where it is defined.
- A row for each type listed in the table, placed following the row in which the type is first mentioned.

Table 1-4 is an example of the request elements and complex types for the exportASR operation.

Table 1-4 Elements and Types for the Request Example

Name	Defined As	Type Description	File Name
exportASRReq	element	exportASR	ASRWebService.wsdl
exportASR	element	exportASR	ASRWebService_schema .xsd
exportASR	complexType	documentNumber	ASRWebService_schema .xsd
documentNumber	int	The order number for export.	ASRWebService_schema .xsd

In this example, exportASRReq is the name value in first row. This exportASR element is defined as the input message element for the operation. It is defined as an element and is located in the **ASRWebService.wsdl** file. (See Example 1-6 for an excerpt of this input message definition.)

The type for the exportASRReq element is exportASR. This exportASR type is then listed in the second row and similarly described.

This pattern continues until one of the following is true:

- The item referenced is list of form fields.
- The item referenced is a series of fields where the series contains simple types or complex types that are located in the same schema file.

To get specific information about fields, you consult the XSD schema file itself. For example, you find the exportASR element in the **ASRWebService_schema.xsd** file.

Similarly, response messages are defined in tables that have the same format and layout as the request information. Refer to the Access Service Ordering Guidelines (ASOG) documentation for field level details and error message information.

Table 1-5 shows an example of the possible error messages for the exportASR operation.

Table 1-5 Error Messages for the exportASR Operation

Error Message	Cause	Resolution
5855 Failed: No Order Found for PON 123.	The document number passed in the input does not exist in database.	Populate a valid document number which needs to be exported.
The order for PON 123 is received not sent and cannot be exported.	The document number provided in the input is a received order and not a sent order.	For an export request, you must provide a sent order and not a received order.

Using the Access Service Ordering Guidelines

The ASR Web Services and its operations are defined based on the Access Service Ordering Guidelines (ASOG). Refer to the guidelines of the desired release to get more information on the field level validations and additional operation details. You can access this documentation at this website:

https://www.atis.org

How to Use Multiple Versions of ASR

Two different ASR versions can be deployed at the same time. To invoke the correct version of the ASR operation, use the release number within the WSDL. Refer to "About WSDL Locations" for more information on the WSDL location URLs.

Deploying, Testing, and Invoking Secure Web Services

This section provides information about deploying, testing, and invoking secure ASR Web Services.

About Deploying ASR Web Services

The ASR Web Services WAR file is part of the ASR EAR file. When the appropriate ASR EAR file is deployed the web services are also deployed. You can deploy ASR and the ASR Web Services in a single server or in a clustered server environment. If you already have the ASR and the ASR Web Services deployed then you will need to first undeploy them before you can deploy again.

See *ASR Installation Guide* for information on deploying and undeploying the ASR EAR file.

About Testing ASR Web Services

You can test the ASR Web Services in several ways, for instance by testing using SoapUI or by writing a Java test client. See *MSS Web Services Developer's Guide* for information on testing.

About Invoking Secure ASR Web Services

The ASR Web Service operations are secured in the same way that MSS Web Services are secured. See *MSS Web Services Developer's Guide* for information on this topic.

Troubleshooting

See *MSS Web Services Developer's Guide* for information on the troubleshooting topic.

Understanding the Differences in CORBA APIs and Web Services

If you currently use the CORBA APIs, this section helps you understand the differences in the CORBA APIs and the web services. Understanding these differences helps you move to the web services and convert from IDL definitions to schema definitions. Most of the APIs have alignment, but the following items detail a few differences:

- The data type fields of MSVDate change to a string data type. The data type format for the date string fields is “yyyy-mm-dd" such as a “2017-07-20" format.

  For the importExternalDLR operation, this change affects the following fields:

  – ISS

  – PTD

  - DD

  For the exportCABS operation, this change affects the following fields:

  - lastModifiedDate
  - inServiceDate
  - serviceTermDate
  - transferDate
  - extractDate
  - vtaTermDate
  - dateOfAgencyAuthorization
  - desiredDueDate

- There are no blank structures for empty forms, except for the TQ form.

  For example, in the CORBA API (not supported now) export of an ASR, the operation returns empty structures for empty forms. For the exportASR web service operation, no empty forms are returned. One exception is for the ASR Web Service where a TQ form returns an empty structure.

- The handling of empty tags in the output data is different in the web services than it is for CORBA APIs (not supported now).

  During an exportASR web service operation, the response structure has:

  - either this definition

    ```
    <tagName></tagName>
    ```

  - or this type of definition

    ```
    <tagName/>
    ```

    where *tagName* is an element or complex type in the schema.

- The easbdw field (in the PIP, DIS and ACI forms) changes to have a unit of measure value included in the web service schemas. An example easbdw field value is “1.45K" which previously was “1.45" without a unit of measure.
- All character data type fields change to a string data type with length of 1 in the schema files.

2

ASR Web Service Reference

This chapter provides information about Oracle Communications MetaSolv Solution (MSS) Access Service Request (ASR) Web Service.

About the ASR Web Service

The ASR Web Service enables an external system to import and export various types of ASR orders in MSS. ASR Web Service operations enable you to:

- Import and export ASR orders with order types such as transport, trunk, SES, Ring, PIP, DIS, and WAL
- Import and export a Confirmation Notice
- Import and export a Clarification Request
- Import and export a Provider Test Acceptance
- Import an external DLR

About the ASR Web Service Packaging

The ASR Web Service is packaged in the ASR EAR file. The EAR file contains the **asrws.war** file. When the installer deploys the EAR file, the ASR Web Service is automatically deployed and ready to use.

About the ASR WSDL, WAR, and Schema Files

The ASR Web Service is defined by the **ASRWebService.wsdl** file and is supported by numerous schema files. The WSDL file and supporting schema files are located in the **asrws.war** file.

See "Understanding ASR Web Services" for more information about WSDL and WAR files, and about their directory locations in the EAR file.

About ASR Schema Files

Several schema files support the ASR Web Service. Within the **asrws.war** file, the schema files are located in the **WEB-INF/wsdls** directory. These schemas are categorized as ASR form schemas, an external DLR schema, and a query ASR schema.

ASR Form Schemas

The ASR form schemas are the schemas that define each ASR form. The following are the ASR form schema files:

- ACI.xsd
- ASR.xsd
- ASRWebService_schema.xsd

- CN.xsd
- CR.xsd
- DIS.xsd
- EOD.xsd
- EUSA.xsd
- EVC.xsd
- FGA.xsd
- MSL.xsd
- MULTI-EC.xsd
- NAI.xsd
- PC.xsd
- PIP.xsd
- PTA.xsd
- RING-ARI.xsd
- SALI.xsd
- SES.xsd
- TQ.xsd
- TRANSPORT.xsd
- TRUNKING.xsd
- VC.xsd
- VCAT.xsd
- WAL.xsd

External DLR Schema

The external DLR schema contains types for the import external DLR web service operation.

The external DLR schema is the following file:

- externalDLR.xsd

Query ASR Schema

The query ASR schema contains types for the query ASR by criteria web service operation.

The query ASR schema is the following file:

- queryASRByCriteria.xsd

importCN Operation

The importCN operation enables you to import a confirmation notice type ASR order. This operation takes the confirmation notice information as input and returns a success message in the response.

The following are the request and response structures:

- **Request Structure:** importCNReq
- **Response Structure:** importCNReqResponse

importCNReq

The importCNReq element contains the input information for the operation. Each row in Table 2-1 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-1 Payload Information for the Request

Name	Defined As	Type Definition	File Name
importCNReq	element	importCN	ASRWebService.wsdl
importCN	element	importCN	ASRWebService_schema.xsd
importCN	complexType	Complex type with a Cn	ASRWebService_schema.xsd
aCn	element	cn	ASRWebService_schema.xsd
cn	complexType	Complex type with a list of CN form fields	CN.xsd
metaSolvUserDataValue	complexType	Refer Table 2-39	ASRWebService_schema.xsd

The details of the required fields in each form can be found in the ASOG documentation.

importCNReqResponse

The importCNReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-2 describes the returned information in the response.

Table 2-2 Payload Information for the Response

Name	Defined As	Type Definition	File Name
importCNReqResponse	element	importCNResponse	ASRWebService.wsdl
importCNResponse	element	importCNResponse	ASRWebService_schema.xsd
importCNResponse	complexType	Complex type with a result	ASRWebService_schema.xsd
result	string	Contains the string “success" for successful output.	ASRWebService_schema.xsd

Refer to the ASOG documentation for the error messages given by the importCN operation. This documentation contains the list of valid errors thrown based on different scenarios.

importCR Operation

The importCR operation enables you to import a clarification request type ASR order. This operation takes the clarification request information as input and returns a success message in the response.

The following are the request and response structures:

- **Request Structure:** importCRReq
- **Response Structure:** importCRReqResponse

importCRReq

The importCRReq element contains the input information for the operation. Each row in Table 2-3 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-3 Payload Information for the Request

Name	Defined As	Type Definition	File Name
importCRReq	element	importCR	ASRWebService.wsdl
importCR	element	importCR	ASRWebService_schema.xsd
importCR	complexType	Complex type with a Cr	ASRWebService_schema.xsd
aCr	element	cr	ASRWebService_schema.xsd
cr	complexType	Complex type with a list of CR form fields	CR.xsd
metaSolvUserDataValue	complexType	Refer Table 2-39	ASRWebService_schema.xsd

The details of the required fields in each form can be found in the ASOG documentation.

importCRReqResponse

The importCRReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-4 describes the returned information in the response.

Table 2-4 Payload Information for the Response

Name	Defined As	Type Definition	File Name
importCRReqResponse	element	importCRResponse	ASRWebService.wsdl
importCRResponse	element	importCRResponse	ASRWebService_schema.xsd
importCRResponse	complexType	Complex type with a result	ASRWebService_schema.xsd

Table 2-4 (Cont.) Payload Information for the Response

Name	Defined As	Type Definition	File Name
result	string	Contains the string “success" for successful output.	ASRWebService_schema.xsd

Refer to the ASOG documentation for the error messages given by the importCR operation. This documentation contains the list of valid errors thrown based on different scenarios.

importExternalDLR Operation

The importExternalDLR operation enables you to import external DLR information for ASR orders. The input for the operation is DLR related information and the operation provides a successful completion message as the output.

The following are the request and response structures:

- **Request Structure:** importExternalDLRReq
- **Response Structure:** importExternalDLRReqResponse

importExternalDLRReq

The importExternalDLRReq element contains the input information for the operation. Each row in Table 2-5 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-5 Payload Information for the Request

Name	Defined As	Type Description	File Name
importExternalDLRReq	element	importExternalDLR	ASRWebService.wsdl
importExternalDLR	element	importExternalDLR	ASRWebService_schema.xsd
importExternalDLR	complexType	Contains the aExternalDLR element	ASRWebService_schema.xsd
aExternalDLR	element	externalDLR	ASRWebService_schema.xsd
externalDLR	complexType	Complex type with the following: • externalDLRAdminInfo • externalDLRDesignInfo • externalDLREndUserInfo • externalDLRDesignLines • externalDLRNotes	externalDLR.xsd
externalDLRAdminInfo	complexType	Complex type with a list of fields	externalDLR.xsd
externalDLRDesignInfo	complexType	Complex type with a list of fields	externalDLR.xsd
externalDLREndUserInfo	complexType	Complex type with a list of fields	externalDLR.xsd
externalDLRDesignLines	complexType	Complex type with a list of fields	externalDLR.xsd
externalDLRNotes	complexType	Complex type with a list of fields	externalDLR.xsd

The details of the required fields can be found in the ASOG documentation.

importExternalDLRReqResponse

The importExternalDLRReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-6 describes the returned information in the response.

Table 2-6 Payload Information for the Response

Name	Defined As	Type Description	File Name
importExternalDLRReqResponse	element	importExternalDLRResponse	ASRWebService.wsdl
importExternalDLRResponse	element	importExternalDLRResponse	ASRWebService_schema.xsd
importExternalDLRResponse	complexType	Complex type with a result	ASRWebService_schema.xsd
result	string	Contains the string “success" for successful output.	ASRWebService_schema.xsd

Refer to the ASOG documentation for the error messages given by the importExternalDLR operation. This documentation contains the list of valid errors thrown based on different scenarios.

importDIS Operation

The importDIS operation enables you to import DIS type ASR orders. This operation takes the DIS related form information as input and provides the document number in the response.

The following are the request and response structures:

- **Request Structure:** importDISReq
- **Response Structure:** importDISReqResponse

importDISReq

The importDISReq element contains the input information for the operation. Each row in Table 2-7 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-7 Payload Information for the Request

Name	Defined As	Type Definition	File Name
importDISReq	element	importDIS	ASRWebService.wsdl
importDIS	element	importDIS	ASRWebService_schema.xsd
importDIS	complexType	Complex type with ASR	ASRWebService_schema.xsd
ASR	element	disASR	ASRWebService_schema.xsd

Table 2-7 (Cont.) Payload Information for the Request

Name	Defined As	Type Definition	File Name
disASR	complexType	Contains a list of forms: • aASR • aDISSeq • aMulti_ECSeq • aACISeq • aSALISeq	ASRWebService_schema.xsd
aASR	element	asr	ASRWebService_schema.xsd
asr	complexType	Contains the list of ASR form fields	ASR.xsd
aDISSeq	element	dis	ASRWebService_schema.xsd
dis	complexType	Contains the list of DIS form fields	DIS.xsd
aMulti_ECSeq	element	multiEC	ASRWebService_schema.xsd
multiEC	complexType	Contains the list of multi EC form fields	MULTI-EC.xsd
aACISeq	element	aci	ASRWebService_schema.xsd
aci	complexType	Contains the list of ACI form fields	ACI.xsd
aSALISeq	element	sali	ASRWebService_schema.xsd
sali	complexType	Contains the list of SALI form fields	SALI.xsd
metaSolvUserDataValue	complexType	Refer Table 2-39	ASRWebService_schema.xsd

The details of the required fields in each form can be found in the ASOG guidelines document.

importDISReqResponse

The importDISReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-8 describes the returned information in the response.

Table 2-8 Payload Information for the Response

Name	Defined As	Type Definition	File Name
importDISReqResponse	element	importDISResponse	ASRWebService.wsdl
importDISResponse	element	importDISResponse	ASRWebService_schema.xsd
importDISResponse	complexType	Complex type with a document number	ASRWebService_schema.xsd
documentNumber	int	The newly created document number	ASRWebService_schema.xsd

Refer to the ASOG documentation for the error messages given by the importDIS operation. This documentation contains the list of valid errors thrown based on different scenarios.

importEUSA Operation

The importEUSA operation enables you to import EUSA type ASR orders. This operation takes the EUSA related form information as input and provides the document number in the response.

The following are the request and response structures:

- **Request Structure:** importEUSAReq
- **Response Structure:** importEUSAReqResponse

importEUSAReq

The importEUSAReq element contains the input information for the operation. Each row in Table 2-9 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-9 Payload Information for the Request

Name	Defined As	Type Definition	File Name
importEUSAReq	element	importEUSA	ASRWebService.wsdl
importEUSA	element	importEUSA	ASRWebService_schema.xsd
importEUSA	complexType	eusaASR	ASRWebService_schema.xsd
eusaASR	complexType	Contains a list of forms: • aASR • aEUSA • aMulti_ECSeq • aACISeq • aNAISeq • aMSLSeq • aVCSeq • aSALISeq • aPCSeq • aEVCSeq • aVCATSeq	ASRWebService_schema.xsd
aASR	element	asr	ASRWebService_schema.xsd
asr	complexType	Contains the list of ASR form fields	ASR.xsd
aEUSA	element	eusa	ASRWebService_schema.xsd
eusa	complexType	Contains the list of EUSA form fields	EUSA.xsd
aSALISeq	element	sali	ASRWebService_schema.xsd
sali	complexType	Contains the list of SALI form fields	SALI.xsd
aMulti_ECSeq	element	multiEC	ASRWebService_schema.xsd
multiEC	complexType	Contains the list of multi EC form fields	MULTI-EC.xsd

Table 2-9 (Cont.) Payload Information for the Request

Name	Defined As	Type Definition	File Name
aACISeq	element	aci	ASRWebService_schema.xsd
aci	complexType	Contains the list of ACI form fields	ACI.xsd
aNAI	element	nai	ASRWebService_schema.xsd
nai	complexType	Contains the list of NAI form fields	NAI.xsd
aMSLSeq	element	msl	ASRWebService_schema.xsd
msl	complexType	Contains the list of MSL form fields	MSL.xsd
aVCSeq	element	vc	ASRWebService_schema.xsd
vc	complexType	Contains the list of VC form fields	VC.xsd
aPCSeq	element	pc	ASRWebService_schema.xsd
pc	complexType	Contains the list of PC form fields	PC.xsd
aEVCSeq	element	evc	ASRWebService_schema.xsd
evc	complexType	Contains the list of EVC form fields	EVC.xsd
aVCATSeq	element	vcat	ASRWebService_schema.xsd
vcat	complexType	Contains the list of VCAT form fields	VCAT.xsd
metaSolvUserDataValue	complexType	Refer Table 2-39	ASRWebService_schema.xsd

The details of the required fields in each form can be found in the ASOG documentation.

importEUSAReqResponse

The importEUSAReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-10 describes the returned information in the response.

Table 2-10 Payload Information for the Response

Name	Defined As	Type Definition	File Name
importEUSAReqResponse	element	importEUSAResponse	ASRWebSerivce.wsdl
importEUSAResponse	element	importEUSAResponse	ASRWebService_schema.xsd
importEUSAResponse	complexType	Complex type with a document number	ASRWebService_schema.xsd
documentNumber	int	The newly created document number	ASRWebService_schema.xsd

Refer to the ASOG documentation for the error messages given by the importEUSA operation. This documentation contains the list of valid errors thrown based on different scenarios.

importFGA Operation

The importFGA operation enables you to import an FGA request type ASR order. This operation takes the FGA related form details as input and provides document number as output.

The following are the request and response structures:

- **Request Structure:** importFGAReq
- **Response Structure:** importFGAReqResponse

importFGAReq

The importFGAReq element contains the input information for the operation. Each row in Table 2-11 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-11 Payload Information for the Request

Name	Defined As	Type Definition	File Name
importFGAReq	element	importFGA	ASRWebService.wsdl
importFGA	element	importFGA	ASRWebService_schema.xsd
importFGA	complexType	Contains a sequence of the ASR element	ASRWebService_schema.xsd
ASR	element	fgaASR	ASRWebService_schema.xsd
fgaASR	complexType	Contains the list of forms • aASR • aTransportSeq • aFGA • aMulti_ECSeq • aACISeq • aNAISeq • aMSLSeq • aSALISeq	ASRWebService_schema.xsd
aASR	element	asr	ASRWebService_schema.xsd
asr	complexType	Contains the list of ASR form fields	ASR.xsd
aTransportSeq	element	transport	ASRWebService_schema.xsd
transport	complexType	Contains the list of Transport form fields	TRANSPORT.xsd
aFGA	element	fga	ASRWebService_schema.xsd
fga	complexType	Contains the list of FGA form fields	FGA.xsd
aMulti_ECSeq	element	multiEC	ASRWebService_schema.xsd

Table 2-11 (Cont.) Payload Information for the Request

Name	Defined As	Type Definition	File Name
multiEC	complexType	Contains the list of multi EC form fields	MULTI-EC.xsd
aACISeq	element	aci	ASRWebService_schema.xsd
aci	complexType	Contains the list of ACI form fields	ACI.xsd
aNAI	element	nai	ASRWebService_schema.xsd
nai	complexType	Contains the list of NAI form fields	NAI.xsd
aMSLSeq	element	msl	ASRWebService_schema.xsd
msl	complexType	Contains the list of MSL form fields	MSL.xsd
aSALISeq	element	sali	ASRWebService_schema.xsd
sali	complexType	Contains the list of SALI form fields	SALI.xsd
metaSolvUserDataValue	complexType	Refer Table 2-39	ASRWebService_schema.xsd

The details of the required fields in each form can be found in the ASOG documentation.

importFGAReqResponse

The importFGAReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-12 describes the returned information in the response.

Table 2-12 Payload Information for the Response

Name	Defined As	Type Description	File Name
importFGAReqResponse	element	importFGAResponse	ASRWebService.wsdl
importFGAResponse	element	importFGAResponse	ASRWebService_schema.xsd
importFGAResponse	complexType	Complex type with a document number	ASRWebService_schema.xsd
documentNumber	int	The newly created document number	ASRWebService_schema.xsd

Refer to the ASOG documentation for the error messages given by the importFGA operation. This documentation contains the list of valid errors thrown based on different scenarios.

importPIP Operation

The importPIP operation enables you to import PIP request type ASR orders. This operation takes the PIP related form information as input and provides the document number in the response.

The following are the request and response structures:

- **Request Structure:** importPIPReq
- **Response Structure:** importPIPReqResponse

importPIPReq

The importPIPReq element contains the input information for the operation. Each row in Table 2-13 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-13 Payload Information for the Request

Name	Defined As	Type Definition	File Name
importPIPReq	element	importPIP	ASRWebService.wsdl
importPIP	element	importPIP	ASRWebService_schema.xsd
importPIP	complexType	Complex type with ASR	ASRWebService_schema.xsd
ASR	element	pipASR	ASRWebService_schema.xsd
pipASR	complexType	Contains a list of forms: • aASR • aPIPSeq • aSALISeq • aMulti_ECSeq • aACISeq • aPVCSeq • aEVCSeq	ASRWebService_schema.xsd
aASR	element	asr	ASRWebService_schema.xsd
asr	complexType	Contains the list of ASR form fields	ASR.xsd
aPIPSeq	element	pip	ASRWebService_schema.xsd
pip	complexType	Contains the list of PIP form fields	PIP.xsd
aSALISeq	element	sali	ASRWebService_schema.xsd
sali	complexType	Contains the list of SALI form fields	SALI.xsd
aMulti_ECSeq	element	multiEC	ASRWebService_schema.xsd
multiEC	complexType	Contains the list of multi EC form fields	MULTI-EC.xsd
aACISeq	element	aci	ASRWebService_schema.xsd
aci	complexType	Contains the list of ACI form fields	ACI.xsd
aPVCSeq	element	pvc	ASRWebService_schema.xsd
pvc	complexType	Contains the list of PVC form fields	PIP.xsd
aEVCSeq	element	evc	ASRWebService_schema.xsd
evc	complexType	Contains the list of EVC form fields	EVC.xsd

Table 2-13 (Cont.) Payload Information for the Request

Name	Defined As	Type Definition	File Name
metaSolvUserDataValue	complexType	Refer Table 2-39	ASRWebService_schema.xsd

The details of the required fields in each form can be found in the ASOG documentation.

importPIPReqResponse

The importPIPReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-14 describes the returned information in the response.

Table 2-14 Payload Information for the Response

Name	Defined As	Type Definition	File Name
importPIPReqResponse	element	importPIPResponse	ASRWebService.wsdl
importPIPResponse	element	importPIPResponse	ASRWebService_schema.xsd
importPIPResponse	complexType	Complex type with a document number	ASRWebService_schema.xsd
documentNumber	int	The newly created document number	ASRWebService_schema.xsd

Refer to the ASOG documentation for the error messages given by the importPIP operation. This documentation contains the list of valid errors thrown based on different scenarios.

importPTA Operation

The importPTA operation enables you to import a provider test acceptance type ASR order. This operation takes the provider test acceptance information as input and returns a success message in the response.

The following are the request and response structures:

- **Request Structure:** importPTAReq
- **Response Structure:** importPTAReqResponse

importPTAReq

The importPTAReq element contains the input information for the operation. Each row in Table 2-15 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-15 Payload Information for the Request

Name	Defined As	Type Definition	File Name
importPTAReq	element	importPTA	ASRWebService.wsdl

Table 2-15 (Cont.) Payload Information for the Request

Name	Defined As	Type Definition	File Name
importPTA	element	importPTA	ASRWebService_schema.xsd
importPTA	complexType	Complex type with name aPta	ASRWebService_schema.xsd
aPta	element	pta	ASRWebService_schema.xsd
pta	complexType	Complex type with a list of PTA form fields	PTA.xsd
metaSolvUserDataValue	complexType	Refer Table 2-39	ASRWebService_schema.xsd

The details of the required fields in each form can be found in the ASOG documentation.

importPTAReqResponse

The importPTAReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-16 describes the returned information in the response.

Table 2-16 Payload Information for the Response

Name	Defined As	Type Definition	File Name
importPTAReqResponse	element	importPTAResponse	ASRWebService.wsdl
importPTAResponse	element	importPTAResponse	ASRWebService_schema.xsd
importPTAResponse	complexType	Complex type with a result	ASRWebService_schema.xsd
result	string	Contains the string "success" for successful output.	ASRWebService_schema.xsd

Refer to the ASOG documentation for the error messages given by the importPTA operation. This documentation contains the list of valid errors thrown based on different scenarios.

importRING Operation

The importRING operation enables you to import RING request type ASR orders. This operation takes the RING related form information as input and provides the document number in the response.

The following are the request and response structures:

- **Request Structure:** importRINGReq
- **Response Structure:** importRINGReqResponse

importRINGReq

The importRINGReq element contains the input information for the operation. Each row in Table 2-17 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-17 Payload Information for the Request

Name	Defined As	Type Definition	File Name
importRINGReq	element	importRING	ASRWebService.wsdl
importRING	element	importRING	ASRWebService_schema.xsd
importRING	complexType	Complex type with ASR	ASRWebService_schema.xsd
ASR	element	ringASR	ASRWebService_schema.xsd
ringASR	complexType	Contains a list of forms: • aASR • aRing • aARISeq • aMulti_ECSeq • aSALISeq • aNAISeq • aPCSeq • aVCATSeq	ASRWebService_schema.xsd
aASR	element	asr	ASRWebService_schema.xsd
asr	complexType	Contains the list of ASR form fields	ASR.xsd
aRing	element	ring	ASRWebService_schema.xsd
ring	complexType	Contains the list of RING form fields	RING-ARI.xsd
aARISeq	element	ari	ASRWebService_schema.xsd
ari	complexType	Contains the list of ARI form fields	RING-ARI.xsd
aMulti_ECSeq	element	multiEC	ASRWebService_schema.xsd
multiEC	complexType	Contains the list of multi EC form fields	MULTI-EC.xsd
aSALISeq	element	sali	ASRWebService_schema.xsd
sali	complexType	Contains the list of SALI form fields	SALI.xsd
aNAI	element	nai	ASRWebService_schema.xsd
nai	complexType	Contains the list of NAI form fields	NAI.xsd
aPCSeq	element	pc	ASRWebService_schema.xsd
pc	complexType	Contains the list of PC form fields	PC.xsd

Table 2-17 (Cont.) Payload Information for the Request

Name	Defined As	Type Definition	File Name
aVCATSeq	element	vcat	ASRWebService_schema.xsd
vcat	complexType	Contains the list of VCAT form fields	VCAT.xsd
metaSolvUserDataValue	complexType	Refer Table 2-39	ASRWebService_schema.xsd

The details of the required fields in each form can be found in the ASOG documentation.

importRINGReqResponse

The importRINGReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-18 describes the returned information in the response.

Table 2-18 Payload Information for the Response

Name	Defined As	Type Definition	File Name
importRINGReqResponse	element	importRINGResponse	ASRWebService.wsdl
importRINGResponse	element	importRINGResponse	ASRWebService_schema.xsd
importRINGResponse	complexType	Complex type with a document number	ASRWebService_schema.xsd
documentNumber	int	The newly created document number	ASRWebService_schema.xsd

Refer to the ASOG documentation for the error messages given by the importRING operation. This documentation contains the list of valid errors thrown based on different scenarios.

importSES Operation

The importSES operation enables you to import SES request type ASR orders. This operation takes the SES related form details as an input and provides document number as an output.

The following are the request and response structures:

- **Request Structure:** importSESReq
- **Response Structure:** importSESReqResponse

importSESReq

The importSESReq element contains the input information for the operation. Each row in Table 2-19 describes the element or type name, the XSD declaration, the type description, and the file name that contains the item's definition.

Table 2-19 Payload Information for the Request

Name	Defined As	Type Definition	File Name
importSESReq	element	importSES	ASRWebService.wsdl
importSES	element	importSES	ASRWebService_schema.xsd
importSES	complexType	Complex Type with ASR	ASRWebService_schema.xsd
ASR	element	sesASR	ASRWebService_schema.xsd
sesASR	complexType	Contains a list of forms: • aASR • aSES • aMulti_ECSeq • aACISeq • aNAISeq • aSALISeq • aEVCSeq	ASRWebService_schema.xsd
aASR	element	asr	ASRWebService_schema.xsd
asr	complexType	Contains the list of ASR form fields	ASR.xsd
aSESSeq	element	ses	ASRWebService_schema.xsd
ses	complexType	Contains the list of SES form fields	SES.xsd
aMulti_ECSeq	element	multiEC	ASRWebService_schema.xsd
multiEC	complexType	Contains the list of multi EC form fields	MULTI-EC.xsd
aACISeq	element	aci	ASRWebService_schema.xsd
aci	complexType	Contains the list of ACI form fields	ACI.xsd
aNAI	element	nai	ASRWebService_schema.xsd
nai	complexType	Contains the list of NAI form fields	NAI.xsd
aSALISeq	element	sali	ASRWebService_schema.xsd
sali	complexType	Contains the list of SALI form fields	SALI.xsd
aEVCSeq	element	evc	ASRWebService_schema.xsd
evc	complexType	Contains the list of EVC form fields	EVC.xsd
metaSolvUserDataValue	complexType	Refer Table 2-39	ASRWebService_schema.xsd

The details of the required fields in each form can be found in the ASOG documentation.

importSESReqResponse

The importSESReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-20 describes the returned information in the response.

Table 2-20 Payload Information for the Response

Name	Defined As	Type Definition	File Name
importSESReqResponse	element	importSESResponse	ASRWebService.wsdl
importSESResponse	element	importSESResponse	ASRWebService_schema.xsd
importSESResponse	complexType	Complex type with a document number	ASRWebService_schema.xsd
documentNumber	int	The newly created document number	ASRWebService_schema.xsd

Refer to the ASOG documentation for the error messages given by the importSES operation. This documentation contains the list of valid errors thrown based on different scenarios.

importTransport Operation

The importTransport operation enables you to import transport type ASR orders. This operation takes the transport related form information as input and provides the document number in the response.

The following are the request and response structures:

- **Request Structure:** importTransportReq
- **Response Structure:** importTransportReqResponse

importTransportReq

The importTransportReq element contains the input information for the operation. Each row in Table 2-21 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-21 Payload Information for the Request

Name	Defined As	Type Definition	File Name
importTransportReq	element	importTransport	ASRWebService.wsdl
importTransport	element	importTransport	ASRWebService_schema.xsd
importTransport	complexType	Complex type with ASR	ASRWebService_schema.xsd
ASR	element	transportASR	ASRWebService_schema.xsd

Table 2-21 (Cont.) Payload Information for the Request

Name	Defined As	Type Definition	File Name
transportASR	complexType	Contains a list of forms: • aASR • aTransportSeq • aMulti_ECSeq • aACISeq • aNAISeq • aMSLSeq • aVCSeq • aSALISeq • aPCSeq • aEVCSeq • aVCATSeq	ASRWebService_schema.xsd
aASR	element	asr	ASRWebService_schema.xsd
asr	complexType	Contains the list of ASR form fields	ASR.xsd
aTransportSeq	element	transport	ASRWebService_schema.xsd
transport	complexType	Contains the list of Transport form fields	TRANSPORT.xsd
aMulti_ECSeq	element	multiEC	ASRWebService_schema.xsd
multiEC	complexType	Contains the list of multi EC form fields	MULTI-EC.xsd
aACISeq	element	aci	ASRWebService_schema.xsd
aci	complexType	Contains the list of ACI form fields	ACI.xsd
aNAI	element	nai	ASRWebService_schema.xsd
nai	complexType	Contains the list of NAI form fields	NAI.xsd
aMSLSeq	element	msl	ASRWebService_schema.xsd
msl	complexType	Contains the list of MSL form fields	MSL.xsd
aVCSeq	element	vc	ASRWebService_schema.xsd
vc	complexType	Contains the list of VC form fields	VC.xsd
aSALISeq	element	sali	ASRWebService_schema.xsd
sali	complexType	Contains the list of SALI form fields	SALI.xsd
aPCSeq	element	pc	ASRWebService_schema.xsd
pc	complexType	Contains the list of PC form fields	PC.xsd
aEVCSeq	element	evc	ASRWebService_schema.xsd
evc	complexType	Contains the list of EVC form fields	EVC.xsd

Table 2-21 (Cont.) Payload Information for the Request

Name	Defined As	Type Definition	File Name
aVCATSeq	element	vcat	ASRWebService_schema.xsd
vcat	complexType	Contains the list of VCAT form fields	VCAT.xsd
metaSolvUserDataValue	complexType	Refer Table 2-39	ASRWebService_schema.xsd

The details of the required fields in each form can be found in the ASOG documentation.

importTransportReqResponse

The importTransportReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-22 describes the returned information in the response.

Table 2-22 Payload Information for the Response

Name	Defined As	Type Definition	File Name
importTransportReqResponse	element	importTransportResponse	ASRWebService.wsdl
importTransportResponse	element	importTransportResponse	ASRWebService_schema.xsd
importTransportResponse	complexType	Complex type with a document number	ASRWebService_schema.xsd
documentNumber	int	The newly created document number	ASRWebService_schema.xsd

Refer to the ASOG documentation for the error messages given by the importTransport operation. This documentation contains the list of valid errors thrown based on different scenarios.

importTrunking Operation

The importTrunking operation enables you to import trunking type ASR orders. This operation takes the trunking related form information as input and provides the document number in the response.

The following are the request and response structures:

- **Request Structure:** importTrunkingReq
- **Response Structure:** importTrunkingReqResponse

importTrunkingReq

The importTrunkingReq element contains the input information for the operation. Each row in Table 2-23 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-23 Payload Information for the Request

Name	Defined As	Type Definition	File Name
importTrunkingReq	element	importTrunking	ASRWebService.wsdl
importTrunking	element	importTrunking	ASRWebService_schema.xsd
importTrunking	complexType	Complex type with ASR	ASRWebService_schema.xsd
ASR	element	trunkingASR	ASRWebService_schema.xsd
transportASR	complexType	Contains a list of forms: • aASR • aTrunking • aMulti_ECSeq • aACISeq • aTQSeq • aEODSeq • aNAIeq • aPCSeq	ASRWebService_schema.xsd
aASR	element	asr	ASRWebService_schema.xsd
asr	complexType	Contains the list of ASR form fields	ASR.xsd
aTrunking	element	trunking	ASRWebService_schema.xsd
trunking	complexType	Contains the list of trunking form fields	TRANSPORT.xsd
aMulti_ECSeq	element	multiEC	ASRWebService_schema.xsd
multiEC	complexType	Contains the list of multi EC form fields	MULTI-EC.xsd
aACISeq	element	aci	ASRWebService_schema.xsd
aci	complexType	Contains the list of ACI form fields	ACI.xsd
aTQSeq	element	tq	ASRWebService_schema.xsd
tq	complexType	Contains the list of TQ form fields	TQ.xsd
aEODSeq	element	eod	ASRWebService_schema.xsd
eod	complexType	Contains the list of EOD form fields	EOD.xsd
aNAI	element	nai	ASRWebService_schema.xsd
nai	complexType	Contains the list of NAI form fields	NAI.xsd
aPCSeq	element	pc	ASRWebService_schema.xsd
pc	complexType	Contains the list of PC form fields	PC.xsd
metaSolvUserDataValue	complexType	Refer Table 2-39	ASRWebService_schema.xsd

The details of the required fields in each form can be found in the ASOG documentation.

importTrunkingReqResponse

The importTrunkingReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-24 describes the returned information in the response.

Table 2-24 Payload Information for the Response

Name	Defined As	Type Definition	File Name
importTrunkingReqResponse	element	importTrunkingResponse	ASRWebService.wsdl
importTrunkingResponse	element	importTrunkingResponse	ASRWebService_schema.xsd
importTrunkingResponse	complexType	Complex type with a document number	ASRWebService_schema.xsd
documentNumber	int	The newly created document number	ASRWebService_schema.xsd

Refer to the ASOG documentation for the error messages given by the importTrunking operation. This documentation contains the list of valid errors thrown based on different scenarios.

importWAL Operation

The importWAL operation enables you to import WAL type ASR orders. This operation takes the WAL related form information as input and provides the document number in the response.

The following are the request and response structures:

- **Request Structure:** importWALReq
- **Response Structure:** importWALReqResponse

importWALReq

The importWALReq element contains the input information for the operation. Each row in Table 2-25 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-25 Payload Information for the Request

Name	Defined As	Type Definition	File Name
importWALReq	element	importWAL	ASRWebService.wsdl
importWAL	element	importWAL	ASRWebService_schema.xsd
importWAL	complexType	Complex type with ASR	ASRWebService_schema.xsd
ASR	element	walASR	ASRWebService_schema.xsd

Table 2-25 (Cont.) Payload Information for the Request

Name	Defined As	Type Definition	File Name
walASR	complexType	Contains a list of forms: • aASR • aWAL • aMulti_ECSeq • aSALISeq • aACISeq • aMSLSeq	ASRWebService_schema.xsd
aASR	element	asr	ASRWebService_schema.xsd
asr	complexType	Contains the list of ASR form fields	ASR.xsd
aWAL	element	wal	ASRWebService_schema.xsd
wal	complexType	Contains the list of WAL form fields	WAL.xsd
aMulti_ECSeq	element	multiEC	ASRWebService_schema.xsd
multiEC	complexType	Contains the list of multi EC form fields	MULTI-EC.xsd
aSALISeq	element	sali	ASRWebService_schema.xsd
sali	complexType	Contains the list of SALI form fields	SALI.xsd
aACISeq	element	aci	ASRWebService_schema.xsd
aci	complexType	Contains the list of ACI form fields	ACI.xsd
aMSLSeq	element	msl	ASRWebService_schema.xsd
msl	complexType	Contains the list of MSL form fields	MSL.xsd
metaSolvUserDataValue	complexType	Refer Table 2-39	ASRWebService_schema.xsd

The details of the required fields in each form can be found in the ASOG documentation.

importWALReqResponse

The importWALReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-26 describes the returned information in the response.

Table 2-26 Payload Information for the Response

Name	Defined As	Type Definition	File Name
importWALReqResponse	element	importWALResponse	ASRWebService.wsdl
importWALResponse	element	importWALResponse	ASRWebService_schema.xsd
importWALResponse	complexType	Complex type with a document number	ASRWebService_schema.xsd

Table 2-26 (Cont.) Payload Information for the Response

Name	Defined As	Type Definition	File Name
documentNumber	int	The newly created document number	ASRWebService_schema.xsd

Refer to the ASOG documentation for the error messages given by the importWAL operation. This documentation contains the list of valid errors thrown based on different scenarios.

exportASR Operation

The exportASR operation enables you to export ASR order of all types. This operation takes the document number as input and retrieves the details of the ASR order. The requested ASR order must be a **sent** order. If it is **received** order, then the request returns an error.

The following are the request and response structures:

- **Request Structure:** exportASRReq
- **Response Structure:** exportASRReqResponse

exportASRReq

The exportASRReq element contains the input information for the operation. Each row in Table 2-27 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-27 Payload Information for the Request

Name	Defined As	Type Description	File Name
exportASRReq	element	exportASR	ASRWebService.wsdl
exportASR	element	exportASR	ASRWebService_schema.xsd
exportASR	complexType	Complex type with the document number	ASRWebService_schema.xsd
documentNumber	int	The document number to export	ASRWebService_schema.xsd

Required Fields for exportASR

The following tables describe the required fields for the exportASR operation.

Table 2-28 describes the required fields for exportASRReq.

Table 2-28 Required Fields for exportASRReq

Field Name	Data Type	Field Description
documentNumber	int	Document number or order number of the ASR order for the export.

exportASRReqResponse

The exportASRReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-29 describes the returned information in the response.

Table 2-29 Payload Information for the Response

Name	Defined As	Type Description	File Name
exportASRReqResponse	element	exportASRResponse	ASRWebService.wsdl
exportASRResponse	element	expASR	ASRWebService_schema.xsd
expASR	complexType	Contains a list of forms	ASRWebService_schema.xsd
fgaASR	complexType	Contains the list of FGA forms	ASRWebService_schema.xsd
transportASR	complexType	Contains the list of Transport forms	ASRWebService_schema.xsd
eusaASR	complexType	Contains the list of EUSA forms	ASRWebService_schema.xsd
sesASR	complexType	Contains the list of SES forms	ASRWebService_schema.xsd
trunkingASR	complexType	Contains the list of Trunking forms	ASRWebService_schema.xsd
ringASR	complexType	Contains the list of Ring forms	ASRWebService_schema.xsd
walASR	complexType	Contains the list of WAL forms	ASRWebService_schema.xsd
disASR	complexType	Contains the list of DIS forms	ASRWebService_schema.xsd
pipASR	complexType	Contains the list of PIP forms	ASRWebService_schema.xsd
metaSolvUserDataValue	complexType	Refer Table 2-39	ASRWebService_schema.xsd

Table 2-30 describes the error messages for the operation.

Table 2-30 Error Messages for the Operation

Error Message	Cause	Resolution
5855 Failed: No Order Found for PON 123.	The document number passed in the input does not exist in database.	Populate a valid document number which needs to be exported.
The order for PON 123 is received not sent and cannot be exported.	The document number provided in the input is a RECEIVED order and not a SENT order.	For an export request, you must provide a SENT order and not a RECEIVED order.

Refer to the ASOG documentation for additional error messages given by the exportASR operation. This documentation contains the list of valid errors thrown based on different import scenarios.

exportCN Operation

The exportCN operation enables you to export the confirmation notice details of the provided ASR order. This operation takes the document number as input and provides the confirmation notice details as the output.

The following are the request and response structures:

- **Request Structure:** exportCNReq
- **Response Structure:** exportCNReqResponse

exportCNReq

The exportCNReq element contains the input information for the operation. Each row in Table 2-31 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-31 Payload Information for the Request

Name	Defined As	Type Description	File Name
exportCNReq	element	exportCN	ASRWebService.wsdl
exportCN	element	exportCN	ASRWebService_schema.xsd
exportCN	complexType	Complex type with the document number	ASRWebService_schema.xsd
documentNumber	int	The document number to export	ASRWebService_schema.xsd

The details of the required fields in each form can be found in the ASOG documentation.

exportCNReqResponse

The exportCNReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-32 describes the returned information in the response.

Table 2-32 Payload Information for the Response

Name	Defined As	Type Description	File Name
exportCNReqResponse	element	exportCNResponse	ASRWebService.wsdl
exportCNResponse	element	exportCNResponse	ASRWebService_schema.xsd
exportCNResponse	complexType	Complex type with aCn.	ASRWebService_schema.xsd
aCn	element	cn	ASRWebService_schema.xsd
cn	complexType	Complex type with a list of CN form fields	CN.xsd

Table 2-32 (Cont.) Payload Information for the Response

Name	Defined As	Type Description	File Name
metaSolvUserDataValue	complexType	Refer Table 2-39	ASRWebService_schema.xsd

Refer to the ASOG documentation for the error messages given by the exportCN operation. This documentation contains the list of valid errors thrown based on different scenarios.

exportCR Operation

The exportCR operation enables you to export the clarification request details of the provided ASR order. This operation takes the document number as input and provides the clarification request details as the output.

The following are the request and response structures:

- **Request Structure:** exportCRReq
- **Response Structure:** exportCRReqResponse

exportCRReq

The exportCRReq element contains the input information for the operation. Each row in Table 2-33 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-33 Payload Information for the Request

Name	Defined As	Type Description	File Name
exportCRReq	element	exportCR	ASRWebService.wsdl
exportCR	element	exportCR	ASRWebService_schema.xsd
exportCR	complexType	Complex type with the document number	ASRWebService_schema.xsd
documentNumber	int	The document number to export	ASRWebService_schema.xsd

The details of the required fields in each form can be found in the ASOG documentation.

exportCRReqResponse

The exportCRReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-34 describes the returned information in the response.

Table 2-34 Payload Information for the Response

Name	Defined As	Type Description	File Name
exportCRReqResponse	element	exportCRResponse	ASRWebService.wsdl

Table 2-34 (Cont.) Payload Information for the Response

Name	Defined As	Type Description	File Name
exportCRResponse	element	exportCRResponse	ASRWebService_schema.xsd
exportCRResponse	complexType	Complex type with aCr	ASRWebService_schema.xsd
aCr	element	cr	ASRWebService_schema.xsd
cr	complexType	Complex type with a list of CR form fields	CR.xsd
metaSolvUserDataValue	complexType	Refer Table 2-39	ASRWebService_schema.xsd

Refer to the ASOG documentation for the error messages given by the exportCR operation. This documentation contains the list of valid errors thrown based on different scenarios.

exportPTA Operation

The exportPTA operation enables you to export the Provider Test Acceptance details of the provided ASR order. This operation takes the document number as input and provides the provider test acceptance details as the output.

The following are the request and response structures:

- **Request Structure:** exportPTAReq
- **Response Structure:** exportPTAReqResponse

exportPTAReq

The exportPTAReq element contains the input information for the operation. Each row in Table 2-35 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-35 Payload Information for the Request

Name	Defined As	Type Description	File Name
exportPTAReq	element	exportPTA	ASRWebService.wsdl
exportPTA	element	exportPTA	ASRWebService_schema.xsd
exportPTA	complexType	Complex type with the document number	ASRWebService_schema.xsd
documentNumber	int	The document number to export	ASRWebService_schema.xsd

The details of the required fields in each form can be found in the ASOG documentation.

exportPTAReqResponse

The exportPTAReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-36 describes the returned information in the response.

Table 2-36 Payload Information for the Response

Name	Defined As	Type Description	File Name
exportPTAReqResponse	element	exportPTAResponse	ASRWebService.wsdl
exportPTAResponse	element	exportPTAResponse	ASRWebService_schema.xsd
exportPTAResponse	complexType	Complex type with aPta.	ASRWebService_schema.xsd
aPta	element	Pta	ASRWebService_schema.xsd
Pta	complexType	Complex type with a list of PTA form fields	PTA.xsd
metaSolvUserDataValue	complexType	Refer Table 2-39	ASRWebService_schema.xsd

Refer to the ASOG documentation for the error messages given by the exportPTA operation. This documentation contains the list of valid errors thrown based on different scenarios.

queryASRByCriteria Operation

The queryASRByCriteria operation enables you to query various ASR orders. The operation takes the PON, CCNA and ICSC field values as input and provides the ASR order information as the output.

The following are the request and response structures:

- **Request Structure:** queryASRByCriteriaReq
- **Response Structure:** queryASRByCriteriaReqResponse

queryASRByCriteriaReq

The queryASRByCriteriaReq element contains the input information for the operation. Each row in Table 2-37 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 2-37 Payload Information for the Request

Name	Defined As	Type Description	File Name
queryASRByCriteriaReq	element	queryASRByCriteria	ASRWebService.wsdl
queryASRByCriteria	element	queryASRByCriteria	ASRWebService_schema.xsd
queryASRByCriteria	complexType	Complex type with the aQc	ASRWebService_schema.xsd

Table 2-37 (Cont.) Payload Information for the Request

Name	Defined As	Type Description	File Name
aQc	element	queryCriteria	ASRWebService_schema.xsd
queryCriteria	element	Complex type with a set of input fields	queryASRByCriteria.xsd

The details of the required fields in each form can be found in the ASOG documentation.

queryASRByCriteriaReqResponse

The queryASRByCriteriaReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 2-38 describes the returned information in the response.

Table 2-38 Payload Information for the Response

Name	Defined As	Type Description	File Name
queryASRByCriteriaReqResponse	element	queryASRByCriteriaResponse	ASRWebService.wsdl
queryASRByCriteriaResponse	element	queryASRByCriteriaResponse	ASRWebService_schema.xsd
queryASRByCriteriaResponse	complexType	Complex type with aServReqQueryDetailCr	ASRWebService_schema.xsd
aServReqQueryDetail	element	servReqQueryDetail	ASRWebService_schema.xsd
servReqQueryDetail	complexType	Complex type with a list of ASR service request fields	queryASRByCriteria.xsd

Refer to the ASOG documentation for the error messages given by the queryASRByCriteria operation. This documentation contains the list of valid errors thrown based on different scenarios.

metaSolvUserDataValue Definition Information

This section describes the metaSolvUserDataValue definition information.

Table 2-39 metaSolvUserDataValue Definition Information

Name	Defined As	Type Definition	File Name
metaSolvUserDataValue	metaSolvUserDataValue	MetaSolvUserDataValueType	ASRWebService_schema.xsd
MetaSolvUserDataValueType	complexType	UserDataValueType	ASR.xsd
UserDataValueType	complexType	complexType with a list of fields BaseNameValueType	ASR.xsd

Table 2-40 Fields for UserDataValueType

Field Name	Data Type	Field Description
tableNm	string	Table where the user data is stored. This table corresponds to where the user data was originally defined.
keyColumnNm	string	Actual primary key column name of the product area. This is required so the user data row can be associated with one item.
keyValue	integer	Actual value of the product area.

Table 2-41 Fields for BaseNameValueType

Field Name	Data Type	Field Description
column	string	Field is the user data column that is updated.
value	string	Value entered into the defined column.
dataType	string	Function of a user data field. Determines the kind of information a user can enter in a field on the User Data window. Valid values are number, decimal, VARCHAR2, date, dropdown.

3

CABS Web Service Reference

This chapter provides information about Oracle Communications MetaSolv Solution (MSS) Access Service Request (ASR) Carrier Access Billing System (CABS) Web Service.

About the CABS Web Service

The CABS Web Service enables an external system to export and query for billing information contained in the ASR tables in MSS.

- Export CABS information
- Query CABS information with criteria

About the CABS Web Service Packaging

The CABS Web Service is packaged in the ASR EAR file. The EAR file contains the **cabsws.war** file. When the installer deploys the EAR file, the ASR Web Service is automatically deployed and ready to use.

About the CABS WSDL, WAR, and Schema Files

The CABS Web Service is defined by the **CABSWebService.wsdl** file and is supported by schema files. The WSDL file and supporting schema files are located in the **cabsws.war** file.

See "Understanding ASR Web Services" for more information about WSDL and WAR files, and about their directory locations in the EAR file.

About the CABS Schema File

A schema file supports the CABS Web Service. Within **cabsws.war** file, the schema file is located in the **WEB-INF/wsdls** directory. The file **CABSWebService.xsd** is the CABS schema file.

exportCABS Operation

The exportCABS operation enables you to request access billing information for a given document number. Exported data includes general CABS, circuit, location, and USOC information.

The following are the request and response structures:

- **Request Structure:** exportCABSReq
- **Response Structure:** exportCABSReqResponse

exportCABSReq

The exportCABSReq element contains the input information for the operation. Each row in Table 3-1 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 3-1 Payload Information for the Request

Name	Defined As	Type Definition	File Name
exportCABSReq	element	exportCABS	CABSWebService.wsdl
exportCABS	element	exportCABS	CABSWebService.xsd
exportCABS	complexType	Complex type with a document number	CABSWebService.xsd
documentNumber	int	The document number to export	CABSWebService_schema.xsd

Table 3-2 describes the required fields for the exportCABSReq element.

Table 3-2 Required Fields for exportCABS Operation

Field Name	Data Type	Field Description
documentNumber	int	The document number or order number to retrieve the access billing information.

exportCABSReqResponse

The exportCABSReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 3-3 describes the returned information in the response.

Table 3-3 Payload Information for the Response

Name	Defined As	Type Definition	File Name
exportCABSReqResponse	element	exportCABSResponse	CABSWebService.wsdl
exportCABSResponse	element	exportCABSResponse	CABSWebService.xsd
exportCABSResponse	element	cabsData	CABSWebService.xsd
cabsData	complexType	Complex type with a set of fields	CABSWebService.xsd

queryCABSByCriteria Operation

This operation requests a query of CABS by criteria and retrieves the information. Any combination of a CCNA and PON is valid.

The following are the request and response structures:

- **Request Structure:** queryCABSByCriteriaReq
- **Response Structure:** queryCABSByCriteriaReqResponse

queryCABSByCriteriaReq

The queryCABSByCriteriaReq element contains the input information for the operation. Each row in Table 3-4 describes the element or type name, the XSD declaration, the type description, and the file that contains the item's definition.

Table 3-4 Payload Information for the Request

Name	Defined As	Type Definition	File Name
queryCABSByCriteriaReq	element	queryCABSByCriteria	CABSWebService.wsdl
queryCABSByCriteria	element	queryCABSByCriteria	CABSWebService.xsd
queryCABSByCriteria	complexType	Complex type with a CCNA and a PON	CABSWebService.xsd

Table 3-5 describes the required fields for the queryCABSByCriteriaReq element.

Table 3-5 Required Fields for queryCABSByCriteria Operation

Field Name	Data Type	Field Description
CCNA	string	The CCNA value of the document number that you want to query.
PON	string	The PON value of the document number that you want to query.

queryCABSByCriteriaReqResponse

The queryCABSByCriteriaReqResponse element contains the output information for the operation. The information returned in the response indicates if the operation is successful. Table 3-6 describes the returned information in the response.

Table 3-6 Payload Information for the Response

Name	Defined As	Type Definition	File Name
queryCABSByCriteriaReqResponse	element	queryCABSByCriteriaResponse	CABSWebService.wsdl
queryCABSByCriteriaResponse	element	queryCABSByCriteriaResponse	CABSWebService.xsd
queryCABSByCriteriaResponse	element	cabsQueryDetail	CABSWebService.xsd
cabsQueryDetail	complexType	Complex type with a set of fields	CABSWebService.xsd

A

ASR and CABS Schema Changes

This appendix describes the schema changes that have been made to the Access Service Request (ASR) and Carrier Access Billing System (CABS) Web Services to support ASOG 70.

New and Revised ASR Form Schemas

This section provides information about the ASR form schemas that have been added or revised to support the ASR Web Service.

New ASR Form Schemas

There are no new forms in ASR 70.

Revised ASR Form Schemas

Table A-1 lists the revised form schemas in ASR 70.

Table A-1 Revised Form Schemas

Form Schema	Section	Fields
CR.xsd	crErrorDetail	ERROR MESSAGE This field is expanded to 180 alpha/numeric characters.