Oracle® Communications MetaSolv Solution ASR 47

Developer's Reference

July 2013

ORACLE

Oracle Communications MetaSolv Solution ASR 47 Developer’s Reference.

Contents

1

The ASR API

The Oracle Communications MetaSolv Solution Access Service Request (ASR) API is designed to assist you in exchanging Access Services Ordering Guidelines (ASOG) 47 ASRs and related documents with trading partners. The ASR API was designed to satisfy the requirements of the ATIS-0404000-0047 - ATIS-0404025-0047 document. Before designing any software that interfaces with the ASR API, you should familiarize yourself with the contents of those documents, which are available at:

http://www.atis.org

Access Service Request Overview

ASRs are requests—based on industry guidelines—that are used to initiate or change access services. ASRs are submitted by a customer to a provider. Access services include service and facilities provided to originate or terminate InterLATA or Interstate telecommunications. The ASR provides a uniform means of requesting service. Each request contains entries required to order particular service and establish billing of appropriate customer accounts. The request is either for special access or switched access services as specified in the various access service tariffs. The ASR is also a mechanism for ordering local interconnection trunks. Refer to the Ordering and Billing Forum (OBF) documentation for detailed information on the ASR forms and fields.

Implementation Concepts

This section describes the implementation concepts in ASR.

Exporting ASR Forms

The ASR API server assumes that an exported order is OBF compliant. It is the responsibility of the core software product to ensure OBF compliance prior to task generation. The *documentNumber* parameter refers to the MetaSolv Solution document number. All export operations of the ASRSession interface require this parameter. The *referenceNumber* parameter refers to a third-party number. This number is determined by the third party. The purpose of this parameter is to provide the third party with a mechanism for matching a requested ASRSession operation with its resulting notification. All operations of the ASRSession interface, including notification operations, require a reference number.

Importing ASR Forms

The ASR API does not perform exhaustive OBF valid value validations. It is the responsibility of the third-party gateway product to ensure field values are consistent with OBF specifications. The ASR API performs necessary validations to ensure the integrity of the data within the MetaSolv Solution database.

CN—Confirmation Notice

There is one export and import operation for all CNs.

- exportCN
- importCN

CR—Clarification Request

There is one export and import operation for all CRs.

- exportCR
- importCR

MetaSolv Solution ASR Module

An ASR is an order to originate or terminate connections to your company's telephone network. It is submitted by the customer on OBF-developed ASR forms. You use the ASR order entry screens when you enter the customer-submitted service request.

The ASR order entry screens guide you through the steps necessary to correctly enter the order. Each OBF form appears as a section of the ASR order entry screens. The entries you make in each section determine which sections (or forms) are presented. The request type of the order (which you enter on the Administrative Section - Admin 2 window) is one of the main determinants for which forms and sections are presented for an order.

For each ASR, the MetaSolv Solution's ASR module requires you to complete the following administrative sections:

- Administrative
- Billing
- Contact

The ASR module supports the following service-specific forms for ASRs:

- End User Special Access
- Feature Group A
- Ring
- Transport
- Trunking

- WATS Access Line

The ASR module also supports the following forms. These forms accompany service-specific forms when certain services are ordered:

- Additional Circuit Information
- Additional Ring Information
- End Office Detail
- Ethernet Virtual Connection
- Multi-EC
- Multipoint Service Legs
- Network Assignment Information
- Ports Configuration
- Service Address Location Information
- Translation Questionnaire
- Virtual Connection (request types V or X)
- Virtual Concatenation (request types S,E,R,V,X)

The following forms are not specific to ATIS/OBF:

- Circuit Assignment

ASR API IDL Files

These IDL files describe the interfaces your application uses to communicate with the ASR API.

- **WDI.IDL**—This IDL file contains the operations that provide the functionality for the WDIManager interface.
- **WDIASR47.IDL**—This IDL file contains the interfaces and operations exposed by the ASR API.
- **WDIASRTYPES47.IDL**—This IDL file contains data structures for the ASOG 47 forms.

WDIASR.IDL Interfaces

This section describes the WDIASR.IDL interfaces.

ASR API WDIRoot Interface

This interface exposes operations used to connect and disconnect from the MetaSolv Solution ASR API server.

Table 1: WDIRoot Interface Operations

Operation	Description
connect	This operation provides a connection allowing a client to access the ASR API server.
disconnect	This operation disconnects the client from the ASR API server.

ASR API WDIManager Interface

This interface exposes operations that start and destroy an ASRSession, WDISignal, and WDIInSignal.

MetaSolv Solution software manages all database transaction processing for the ASR API. This means MetaSolv Solution establishes the database connection and maintains control of *commit* and *rollback* processing. Consequently, the ASR API does not use the WDITransaction interface.

Table 2: WDIManager Interface Operations

Operation	Description
startASRSession	This operation starts an ASR session for the client using the ASR API server. The returned ASRSession object reference is used to access the ASR API application logic.
destroyASRSession	This operation destroys the ASRSession.
startSignal	This operation starts a Signal for the client using the ASR API server. The returned WDISignal object reference is used to update the status of outbound gateway events.
destroySignal	This operation destroys the Signal.
startInSignal	This operation starts an InSignal, for the client, with the ASR API server. The returned WDIInSignal object reference is used to update the status of inbound gateway events.
destroyInSignal	This operation destroys the InSignal.

ASRSession Interface

This interface exposes operations allowing export and import of ASR orders, confirmations and clarifications.

WDINotification Interface

This interface enables a callback mechanism to notify the client of the result of an operation invoked against the ASR API server. All operations of the ASRSession interface are asynchronous and require a WDINotification object reference.

The *operationFailed* operation is invoked any time an operation of the ASRSession interface fails. This operation notifies the client that the requested operation on the ASRSession interface failed and provides the reason(s) for the failure.

The *importSucceeded* operation is sent any time an import operation of the ASRSession interface succeeds. This operation notifies the client that a requested import operation of the ASRSession interface for the stated reference number completed successfully.

The following table lists the operations in the **WDIASR47.IDL** file with the accompanying notifications.

Table 3: ASR API Operations and Notifications

ASRSession Operation	Related WDINotification Operations
exportASR	*exportEUSASucceeded* *exportFGASucceeded* *exportRINGSucceeded* *exportTRANSPORTSucceeded* *exportTRUNKINGSucceeded* *exportWALSucceeded* *operationFailed*
exportCN	*exportCNSucceeded* *operationFailed*
exportCR	*exportCRSucceeded* *operationFailed*
importCN	*importSucceeded* *operationFailed*
importCR	*importSucceeded* *operationFailed*
importEUSA	*importSucceeded* *operationFailed*

Table 3: ASR API Operations and Notifications

ASRSession Operation	Related WDINotification Operations
importExternalDLR	*importSucceeded* *operationFailed*
importFGA	*importSucceeded* *operationFailed*
importRING	*importSucceeded* *operationFailed*
importTRANSPORT	*importSucceeded* *operationFailed*
importTRUNKING	*importSucceeded* *operationFailed*
importWAL	*importSucceeded* *operationFailed*
queryASRByCriteria	*queryASRByCriteriaSucceeded* *operationFailed*

ASR API Operations

The following are the ASR API operations:

exportASR

This operation requests the export of an ASR for a given *documentNumber*. The WDINotification and *referenceNumber* are used to send the appropriate notification information to the client upon completion of the export.

exportCN

This operation requests the export of a Confirmation Notice for a given *documentNumber*. The WDINotification and *referenceNumber* are used to send the appropriate notification information to the client upon completion of the export.

exportCR

This operation requests the export of a Clarification Request for a given *documentNumber*. The WDINotification and *referenceNumber* are used to send the appropriate notification information to the client upon completion of the export.

importCN

This operation requests the import of an ASOG 47 Confirmation Notice and provides the data for the imported confirmation. The WDINotification and *referenceNumber* are used to send the appropriate notification information to the client upon completion of the import.

importCR

This operation requests the import of an ASOG 47 Clarification Request and provides the data for the imported confirmation. The WDINotification and *referenceNumber* are used to send the appropriate notification information to the client upon completion of the import.

importEUSA

This operation requests the import of an ASOG 47 End User Special Access ASR order and provides the data for the imported order. The WDINotification and *referenceNumber* are used to send the appropriate notification information to the client upon completion of the import.

importExternalDLR

This operation requests the import of an External DLR and provides the data for the imported External DLR. The WDINotification and *referenceNumber* are used to send the appropriate notification information to the client upon completion of the import.

importFGA

This operation requests the import of an ASOG 47 Feature Group A ASR order and provides the data for the imported order. The WDINotification and *referenceNumber* are used to send the appropriate notification information to client upon completion of the import.

importRING

This operation requests the import of an ASOG 47 Ring ASR order and provides the data for the imported order. The WDINotification and *referenceNumber* are used to send the appropriate notification information to the client upon completion of the import.

importTRANSPORT

This operation requests the import of an ASOG 47 Transport ASR order and provides the data for the imported order. The WDINotification and *referenceNumber* are used to send the appropriate notification information to the client upon completion of the import.

importTRUNKING

This operation requests the import of an ASOG 47 Trunking ASR order and provides the data for the imported order. The WDINotification and *referenceNumber* are used to send the appropriate notification information to the client upon completion of the import.

importWAL

This operation requests the import of an ASOG 47 WAL ASR order and provides the data for the order to be imported. The WDINotification and *referenceNumber* are used to send the appropriate notification information to client upon completion of the import.

queryASRByCriteria

This operation allows the query of ASRs by criteria. The query criteria includes any combination of CCNA, PON, and ICSC.

2

The CABS API Operations

The CABS (Carrier Access Billing System) API operations are a subset of the ASR API that allows you to retrieve billing information contained in ASR tables in the MetaSolv Solution database.

There are five structures in the **WDICABSTYPES.IDL** file that hold the following types of information:

- USOC
- Location
- Circuit
- General CABS
- Information query

CABS API IDL Files

The following are the CABS API IDL files:

- **WDI.IDL**—This IDL file contains the operations that provide the functionality for the WDIManager interface.
- **WDIUTIL.IDL**—This IDL file provides functionality common to all MetaSolv Solution APIs.
- **WDICABS.IDL**—This IDL file contains the interfaces and operations exposed by the CABS API.
- **WDICABSTYPES.IDL**—This IDL file contains data structures used in conjunction with the CABS API.

WDICABS.IDL Interfaces

The section describes the WDICABS.IDL interfaces.

CABS API Interfaces Relationship

Figure 1 shows the relationship of the CABS API interfaces.

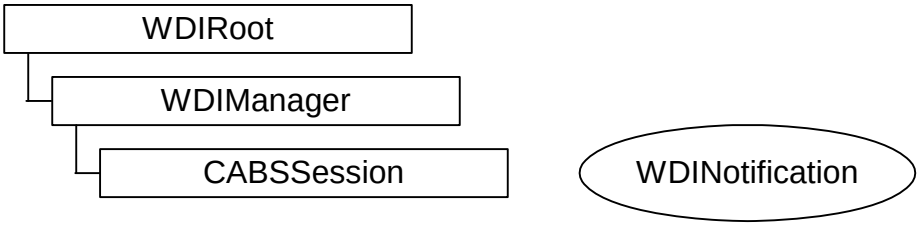

Figure 1: WDICABS.IDL Interfaces

CABS API WDIRoot Interface

This interface exposes operations used to connect and disconnect from the MetaSolv Solution ASR API server.

Table 4 lists the WDIRoot interface operations.

Table 4: WDIRoot Interface Operations

Operation	Description
connect	Returns a reference to WDIManager
disconnect	Terminates the connection

CABS API WDIManager Interface

Table 5 lists the WDIManager interface operations.

Table 5: WDIManager Interface Operations

Operation	Description
startCABSSession	Obtains the object reference of the CABSSession
destroyCABSSession	Terminates the CABSSession
startSignal	Obtains the WDISignal object reference
destroySignal	Terminates the Signal
startInSignal	Obtains the WDI Insignal object reference

Table 5: WDIManager Interface Operations

Operation	Description
destroyInSignal	Terminates the Insignal

CABS API CABSSession Interface Operations

Table 6 lists the CABSSession interface operations.

Table 6: CABSSession Interface Operations

Operation	WDINotification
exportCABS	exportCABSSucceeded operationFailed
queryCABSByCriteria	queryCABSByCriteriaSucceeded operationFailed

CABS API Operations

The following are the CABS API operations:

exportCABS

The *exportCABS* operation requests access billing information for a given document number. Exported data includes general CABS, circuit, location, and USOC information. Refer to the **WDICABSTYPES.IDL** for details on the data structures exported by this operation.

queryCABSByCriteria

This operation requests a query of CABS by criteria and retrieves the information. Any combination of CCNA, and PON can be used. The WDINotification and *referenceNumber* are used to send the appropriate notification information to the client upon completion of the query.

CABS API Structures

These structures are located in the **WDICABSTYPES.IDL** file.

CabCktLocUsoc

The CabCktLocUsoc structure can contain 0 to many rows of data, based on the number of USOCs for a given *documentNumber*. This structure contains USOC information.

CabCktLoc

The CabCktLoc structure can contain 0 to many rows of data, based on the number of circuit locations for a given *documentNumber*. This structure contains location information.

CabCkt

The CabCkt structure can contain 0 to many rows of data, based on the number of circuits for a given *documentNumber*. This structure contains circuit information.

CabsData

The CabsData structure always contains one row of data, based on a given *documentNumber*. This structure contains general CABS information.

CabsQueryDetail

The CabsQueryDetail structure retrieves the following information and makes it available to the user:

- documentNumber
- customerNameAbbreviation
- purchaseOrderNumber
- activityInd
- requestType

A

Appendix A: IDL Changes and Enhancements

This appendix describes the differences between the IDL that supports ASOG 47 and the previous release of the ASR API IDL.

MetaSolv Solution APIs Impacted by IDL Changes and Enhancements

The only IDL changes and enhancements made for ASOG 47 involve the ASR API.

These ASR API IDL files were not changed for ASOG 47:

- **WDI.IDL**
- **WDIUTIL.IDL**
- **WDICABS.IDL**
- **WDICABSTYPES.IDL**

These files were added for ASOG 47:

- **WDIASRTYPES47.IDL**
- **WDIASR47.IDL**

New IDL Files

There are two new IDL files for the ASR API for ASOG 47, **WDIASRTYPES47.IDL** and **WDIASR47.IDL**. These files define all IDL elements of scope MetaSolv::CORBA::ASRTypes47. No IDL elements of other scopes are defined in this file. The **WDIASRTYPES47.IDL** is based on **WDIASRTYPES46.IDL** from the previous release.

WDIASR46.IDL Changed to WDIASR47.IDL

Removed statement to include **WDIASRTYPES46.IDL**.

Added statement to include **WDIASRTYPES47.IDL**.

Revised Data Structures

The following ASR IDL data structures were revised to support ASOG 47:

Fields Added:

struct Transport - sbdw

struct Transport - icol

struct CN - nfrt

struct CNVCDetail - svlan1, svlan2

struct EUSA - sbdw

struct EVCUNIMappingDetail - ei

Fields Modified:

struct ASR - telno (changed field length from 15 to 18)

structASR - initTelno (changed field length from 15 to 18)

structASR - dsgTelno (changed field length from 15 to 18)

struct CR - telno (changed field length from 15 to 18)

struct CNVCDetail - svlan (changed field length from 5 to 10)