Oracle® Communications MetaSolv Solution LSR 9.4

Developer's Reference

April 2011

ORACLE

Oracle Communications MetaSolv Solution LSR 9.4 Developer’s Reference.

Contents

1

The LSR Interconnection API

Overview

Oracle's LSR Interconnection API (LSR API) for LSOG 9 provides IDL for Local Service Ordering Guidelines (LSOG) versions 9.

The LSR API is fully runtime backward compatible because a newer release of the API server continues to work with an existing release of a third-party vendor's product. Some examples are provided below:

- Customer A upgrades to a newer release of the core MetaSolv Solution software and APIs.
- Third-party vendor B is currently deployed, API client product (third-party application B is used by customer A) continues to work against the newer release of the API server. However, third-party vendor B's product does not support any of the new functionality implemented in the newer release of the API server.

  For example, this situation occurs if a new operation is added to an existing interface that uses a new operation on an existing notification interface.

  
  The third-party vendor recompiles with the newer IDL and is required to implement the new operations of existing interfaces using stub implementations. The IDL is not compiler backward compatible.

Beginning with version 4.2.0 the LSR API maintains backward compatibility because each type of LSR order that can be exported has:

- A separate succeeded operation within the WDINotification interface
- A service specific structure in the WDILSRTypes*n* (where *n* represents the LSOG version)

LSR API Interfaces

Figure 1 shows the relationship of the interfaces in the LSR API.

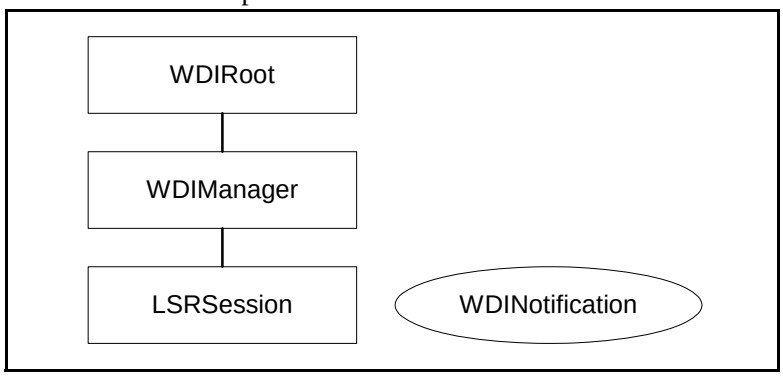

Figure 1: LSRSession API Interfaces

WDIRoot Interface

The following table lists the operations in the WDIRoot interface of the LSR API.

Table 1: WDIRoot Interface Operations

Operation	Description
connect	Obtains the object reference to the WDIManager
disconnect	Terminates the connection

To begin a connection, the third-party application must connect to the MetaSolv Solution Application Server. This connection verifies the user ID and password, and returns the object reference to the API server's WDIRoot. A successful *connect* operation on the WDIRoot object returns an object reference to a WDIManager object.

WDIManager Interface

The LSR API's WDIManager interface enables LSRSession, signal, insignal, and transaction management. Table 2 describes the operations in the WDIManager interface.

Table 2: WDIManager Interface Operations

WDIManager Operation	Description
startLSRSession *destroyLSRSession*	Starts/destroys an LSR session for the client on the LSR API server. StartLSRSession returns an object reference to the LSRSession object.
startTransaction *destroyTransaction*	Starts/destroys a Transaction object for the client on the LSR API server. StartTransaction returns an object reference to the Transaction object. The Transaction object provides the connection to the MetaSolv Solution database. For more information, see "Transaction Management and the LSR API" *on page 3*.
startSignal *destroySignal*	Starts/destroys a WDISignal object for the client on the LSR API server. StartSignal returns a WDISignal object reference that you can use to update the status of outbound gateway events.
startInSignal *destroyInSignal*	Starts/destroys a WDInSignal object for the client on the LSR API server. StartInSignal returns a WDIInSignal object reference that you can use to update the status of inbound gateway events.

Transaction Management and the LSR API

Refer to the Common Architecture chapter of the API Developer's Guide for a complete description of the operations in the table above.

The WDITransaction interface is defined in the WDI.IDL file. For a detailed explanation of transaction and signal processing, see *MetaSolv Solution API Developer's Reference*.

In all cases, you must create and pass a transaction object to the LSR API. After a successful commit, you can reuse the transaction object. After a rollback, the transaction object is destroyed and you must create a new one to continue using the LSR API.

You are required to call the commit or rollback operation after calling the *importDSCN*, *importDSRED*, or *importNPLSR*, operations in all cases. However, there are two possible approaches to handling database transactions when you call the *importLR* and *importLSR* operations. These approaches are:

- Your application can call *commit* and *rollback* in every case. In this case, your application is responsible for ensuring that the data provided in the structures imported into the MetaSolv Solution database via the LSR API is compliant with published OBF/LSOG guidelines.

- The LSR API automatically handles database commits and rollbacks every time you call these operations, with these possible results for any given operation:
  - If the operation fails, the LSR API automatically rolls back the database changes and calls the related failure notification operation.
  - If the operation succeeds, the LSR API automatically commits the database changes and calls the related succeeded notification operation.
  - If the operation succeeds but the *commit* fails, the database automatically rolls back the changes and the LSR API calls the related failure notification operation.

The setting of the ForceAutoCommit parameter in the Session section of the Application Server's GATEWAY.INI file determines the type of transaction management used by the LSR API for the *importLR* and *importLSRCM* operations. Regardless of the setting of this parameter, you are required to call the *commit* or *rollback* operation after calling the *importDSCN*, *importDSRED*, or *importNPLSR* operations.

- If ForceAutoCommit is set to "true" (recommended), the LSR API handles transaction management for these operations.
- If ForceAutoCommit is set to "false", your application handles transaction management for all LSR API operations.

The recommended approach is to have the LSR API handle transaction management unless you must use a single transaction object to group the results of a number of LSR API operations into a unit of work and you want to rollback the entire string of operations if any of the operations fails. When the LSR API handles transaction management for an operation, the commit happens just prior to the API calling the success or failure notification. If you group the results of a number of API operations together in your code but have the LSR API handle transaction management, a number of successful operations could be committed before a later operation fails and your application would be unable to rollback the already committed operations. This could lead to mismatched data or other unexpected results.

When designing your application to handle transaction processing for the purposes of the LSR API, you need to know:

- Until you invoke the *commit* operation, any changes you make are not reflected in the database. If you delay calling *commit*, you can adversely impact the processing of service requests. For example, if you update the status of a gateway event but do not *commit* your update immediately, the gateway event status remains unchanged and no task that is waiting on that event can be completed.
- If you lose your connection to the LSR API for any reason, the LSR API rolls back all transactions for which you have not called *commit*.

Signal Processing and the LSR API

The WDISignal and WDIInSignal interfaces are defined in the WDI.IDL file. These interfaces are used in common by the MetaSolv Solution APIs. For a detailed explanation of signal

processing, see the *MetaSolv Solution API Developer's Reference* for the API version you are using. For the LSR API you need to know:

- The operations in the WDISignal interface allow you to communicate with the LSR API regarding outbound gateway events. Your application is responsible for implementing the *eventOccurred* and *eventTerminated* operations. The LSR API implements this interface's remaining operations.
- The operations in the WDIInSignal interface allow you to communicate with the LSR API regarding inbound gateway events. The LSR API implements all operations in this interface.

LSRSession Interface Operations

The following table lists the operations available in the LSRSession interface of the WDILSR.IDL file.

Table 3: LSRSession Interface Operations

Operation	WDINotifications
exportLSR	*LSRExportLSSucceeded* *LSRExportPSSucceeded* *LSRExportPSLSSucceeded* *LSRExportNPSucceeded* *LSRExportLSNPSucceeded* *LSRExportRSSucceeded* *LSRExportCRSSucceeded* *LSRExportFailed*
exportLR	*LRExportSucceeded* *LRExportFailed*
exportLSRCM	*LSRCMExportSucceeded* *LSRCMExportFailed*
exportDL	*LSRExportDLSucceeded*
importLR	*LRImportSucceeded* *LRImportFailed*
importLSRCM	*LSRCMImportSucceeded* *LSRCMImportFailed*
importNPLSR	*NPLSRImportSucceeded* *NPLSRImportFailed*
importDSRED	*DSREDImportSucceeded* *DSREDImportFailed*

Table 3: LSRSession Interface Operations

Operation	WDINotifications
importDSCN	*DSCNImportSucceeded* *DSCNImportFailed*
listLSR	N/A
listCCNA	N/A
listPONSForCCNA	N/A
ListLSRForPONCCNAVER	N/A

LSRSession Interface Operations

Refer to the current OBF/LSOG guidelines for operation descriptions.

Import Operations

importLR

LSOG 9 Local Service Request Confirmation

importLSRCM

LSOG 9 Local Service Response/Completion

importNPLSR

LSOG 9 Number Portability LSR

importDSRED

LSOG 9 Directory Service Request Error Details

importDSCN

LSOG 9 Directory Service Completion Notice

Export Operations

exportLSR

Every LSR export (*exportLSR)* operation uses the LSR and EU forms. In addition the exported LSR must contain either one of the following mutually exclusive forms: LS, NP, LSNP, RS, CRS, PS or a combination of the PS and LS forms. Table 4 on page 7 lists the forms.

Table 4: LSR Export Form Usage by LSOG Version

Form	LSOG 9
LSR (Local Service Request)	X
EU (End User)	X
LS (Loop Service)	X
NP (Number Portability)	X
LSNP (Loop Service with NP)	X
RS (Resale)	X
CRS (Centrex)	X
PS (Port Service)	X
PS and LS (combined Port and Loop)	X
HGI (Hunt Group Identification)	X

- exportLSRCM—LSOG 9 Local Service Response/Completion
- exportLR—LSOG 9 Local Response
- exportDL—LSOG 9 Directory Listing

List Operations

- listLSR
- listCCNA
- listPONSForCCNA
- listLSRForPONCCNAVER

Process Flows

This section contains sample process flows for each type of signal: solicited and unsolicited. Use the sample flow as a template for developing your own process flows.

Solicited Messages

A solicited message is a message initiated by the MetaSolv Solution software. The MetaSolv Solution software plays the role of the client, and the third-party application plays the role of

the server. The third-party application must use the IDL files provided with Oracle's LSR API to implement the interfaces and operations for the structures shown in the following table:

Table 5: WDIRoot, WDIManager, and WDISignal Operations for Solicited Messages

Interface	For Implementing These Operations
WDIRoot	*connect* *disconnect*
WDIManager	*startSignal* *destroySignal*
WDISignal	*eventOccurred* *eventTerminated*

Sample Solicited Message Process Flow

When the MetaSolv Solution software is the client, the overall process flows as follows:

1. The API client binds to the third-party server to get a WDIRoot object reference.
2. The API client invokes the *connect* operation of the WDIRoot interface, and the *connect* operation yields a WDIManager object reference.
3. The API client invokes the *startSignal* operation of the WDIManager interface to get a WDISignal object reference.
4. The client invokes the *eventOccurred* operation of the WDISignal interface to notify the third-party application that an event registered to them has occurred within the MetaSolv Solution software.
5. The API client invokes the *destroySignal* operation of the WDIManager interface.
6. The API client invokes the *disconnect* operation of the WDIRoot interface.

If the third-party application encounters an error, it throws a WDIExcp as defined by the IDL. The client handles CORBA system exceptions and WDIExcp exceptions.

Unsolicited Messages

An unsolicited message is a message initiated by the third-party application. Oracle's software plays the role of the server, and a third-party application plays the role of the client, with the exception of the callback processing.

Table 6 lists the operation that the LSR API implements. Some IDL compilers require you to provide trivial implementations of these operations in your application.

Table 6: LSR Root, Manager, and Signal Operations for Unsolicited Messages

Interface	Operations Implemented by the LSR API
WDIRoot	*connect* *disconnect*
WDIManager	*startSignal* *destroySignal* *startInSignal* *destroyInSignal* *startTransaction* *destroyTransaction* *startLSRSession* *destroyLSRSession*
WDITransaction	*commit* *rollback*
WDISignal	*eventErrored* *eventInProgress* *eventCompleted*
WDIInSignal	*eventErrored* *eventInProgress* *eventCompleted*
LSRSession	*exportLSR* *exportLR* *exportLSRCM* *exportDL* *importLR* *importLSRCM* *importNPLSR* *importDSRED* *importDSCN* *listLSR* *listCCNA* *listPONSForCCNA* *listLSRForPONCCNAVER*

Table 7 lists the operations that your application must implement.

Table 7: LSR Notification Operations

Interface	Operations Implemented by Your Application
WDINotification	*LSRExportLSSucceeded* *LSRExportPSSucceeded* *LSRExportPSLSSucceeded* *LSRExportNPSucceeded* *LSRExportLSNPSucceeded* *LSRExportRSSucceeded* *LSRExportCRSSucceeded* *LSRExportDLSucceeded* *LSRExportFailed* *LRImportSucceeded* *LRImportFailed* *LSRCMImportSucceeded* *LSRCMImportFailed* *NPLSRImportSucceeded* *NPLSRImportFailed* *DSREDImportSucceeded* *DSREDImportFailed* *DSCNImportSucceeded* *DSCNImportFailed* *LSRImportSucceeded* *LSRImportFailed* *LRExportSucceeded* *LRExportFailed* *LSRCMExportSucceeded* *LSRCMExportFailed*

Sample Business Flows for Sending an LSOG 9 CRS

The following two sample flows show the process for exporting an LSOG 9 CRS (Centrex Resale Services) order and importing an LSOG 9 LR (Local Response).

Exporting an LSOG 9 CRS

Assume the following infrastructure setup is in place:

- A gateway defined pointing to a third-party LSR API server. The host and server names must be those of the third-party's LSR API server.
- The gateway has a gateway event defined for export of the LSR. The export of the LSR is defined as an outbound, order level event.
- A provisioning plan exists with a task that has the export LSR gateway event associated to it.

The following process describes the business flow for exporting or sending an LSOG 9 CRS electronically:

1. The user enters an LSOG 9 LSR for CRS into the MetaSolv Solution software.
2. The user generates tasks for the CRS LSR using the provisioning plan described in the infrastructure setup of this business flow.
3. The user goes to the work queue and initiates the export LSR gateway event on the task. This action causes the following steps to occur:
   - The MetaSolv Solution software binds to the third-party server using the information provided in the definition of the gateway in the MetaSolv Solution software's Infrastructure subsystem. This binding yields a WDIRoot object reference.
   - The MetaSolv Solution software invokes the *connect* operation on the WDIRoot interface yielding a WDIManager object reference.
   - The MetaSolv Solution software invokes the *startSignal* operation of the WDIManager interface yielding a WDISignal object reference.
   - The MetaSolv Solution software invokes the *eventOccurred* operation of the WDISignal interface. The MetaSolv Solution software passes a fully populated WDIEvent structure. This structure informs the third-party that a LSR order is ready for export from the MetaSolv Solution database to the trading partner. Note the ServItemID is null in the WDIEvent structure because the export CRS is an order level event.
4. Upon receiving the *eventOccurred* signal the third-party performs the following:
   - The third-party application binds to the MetaSolv Solution LSR API server to get a WDIRoot object reference.
   - The third-party application invokes the *connect* operation of the WDIRoot interface, which yields a WDIManager object reference.
   - The third-party application invokes the *startTransaction* operation of the WDIManager interface, which yields a WDITransaction object reference. This action starts a database transaction on the MetaSolv Solution LSR API server.
   - The third-party application invokes the *startSignal* operation of the WDIManager interface, which yields a WDISignal object reference. This object reference is used later in the process flow to update Gateway Event status information.
   - The third-party application invokes the *startLSRSession* operation of the WDIManager interface, which yields a LSRSession object reference.
   - The third-party application instantiates a WDINotification object.
   - The third-party application invokes the *exportLSR* operation of the LSRSession interface. The WDITransaction, WDINotification, and WDIEvent.documentNumber are supplied as input parameters.
5. The LSR API server receives and processes the *exportLSR* operation.

6. If the LSR API server is successful it invokes the *LSRExportCRSSucceeded* operation of the WDINotification interface.
7. If the LSR API server is unsuccessful it invokes the *LSRExportFailed* operation of the WDINotification interface.
8. The following process occurs if the *LSRExportCRSSucceeded* operation was invoked:
   - The third-party application invokes the *eventInProgress* operation of the WDISignal interface passing a populated WDIEvent structure. The data to populate the WDIEvent structure comes from the WDIEvent structure passed on the *eventOccurred* operation.
   - The third-party application invokes the *commit* operation of the WDITransaction interface.
   - The third-party application formats the CRS data passed on the notification on to the provider.
   - If the provider successfully processes the CRS:
     - The third-party application invokes the *eventCompleted* operation of the WDISignal interface passing a populated WDIEvent structure. The data to populate the WDIEvent structure comes from the WDIEvent structure passed on the *eventOccurred* operation.
   - If the provider is unable to process the CRS:
     - The third-party application invokes the *eventErrored* operation of the WDISignal interface passing a populated WDIEvent structure. The data to populate the WDIEvent structure comes from the WDIEvent structure passed on the *eventOccurred* operation. The third party can populate the WDIErrSeq sequence with the reason(s) for the failure.
- The third-party application invokes the *commit* operation of the WDITransaction interface.

9. If the *LSRExportFailed* operation was invoked:
   - The third-party application issues the *rollback* operation of the WDITransaction interface.
   - The third-party application invokes the *startTransaction* operation of the WDIManager interface, because the rollback terminates the WDITransaction object reference.
   - The third-party application invokes the *eventErrored* operation of the WDISignal interface passing a populated WDIEvent structure. The data to populate the WDIEvent structure comes from the WDIEvent structure passed on the *eventOccurred* operation. The third-party can populate the WDIErrSeq sequence with the reasons for the failure.
   - The third-party application invokes the *commit* operation of the WDITransaction interface.

10. The third-party application invokes the *destroySignal* operation of the WDIManager interface.
11. The third-party application invokes the *destroyLSRSession* operation of the WDIManager interface.
12. The third-party application invokes the *destroyTransaction* operation on the WDIManager interface.
13. The third-party application invokes the *disconnect* operation of the WDIRoot interface.

Receiving an LSOG 9 LR (Local Response)

Assume the following infrastructure setup is in place:

- A gateway defined pointing to a third-party LSR API server. The host and server names must be those of the third-party's LSR API server.
- The gateway has a gateway event defined for import of the LR. The Import LR should be defined as an inbound, order level event.
- A provisioning plan exists with a task that has the import LR gateway event associated to it.

The following describes the business flow for receiving an LSOG9 LR electronically:

1. The third-party application binds to the LSR API server to get a WDIRoot object reference.
2. The third-party application invokes the *connect* operation of the WDIRoot interface, which yields a WDIManager object reference.
3. The third-party application invokes the *startTransaction* operation of the WDIRoot interface to get a WDITransaction object reference and start a database transaction.
4. The third-party application invokes the *startInSignal* operation of the WDIManager interface, which yields a WDIInSignal object reference.
5. The third-party application invokes the *eventInProgress* operation of the WDIInSignal interface, passing a populated WDIInEvent structure.
6. The third-party application invokes the *commit.*
7. The third-party application invokes the *startLSRSession* operation of the WDIManager interface, which yields a LSRSession object reference.
8. The third-party application instantiates a WDINotification object.
9. The third-party application invokes the *importLR* operation of the LSRSession interface. The WDITransaction, WDINotification, third-party's document identifier, and the WDILSRTypes6 LR structure are supplied as input parameters.

10. The LSR API server processes the *importLR* operation of the LSRSession and invokes the appropriate callback operation of the input WDINotification. In this example the operations are *LRImportSucceeded* and *LRImportFailed*.
11. If the *LRImportSucceeded* operation is invoked, the third party performs the following:
    - The third-party application invokes the *eventCompleted* operation of the WDIInSignal interface.
    - The third-party application invokes the *commit* operation of the WDITransaction object.
12. If the *LRImportFailed* operation was invoked the third party performs the following:
    - Invoke the *rollback* operation of the WDITransaction interface and proceed with the appropriate error logging.
13. The third-party application invokes the *startTransaction* operation of the WDIManager interface, because the rollback terminates the WDITransaction object reference.
14. The third-party application invokes the *eventErrored* operation of the WDIInSignal interface.
15. The third-party application invokes the *commit* operation of the WDITransaction object.
16. The third-party application invokes the *destroyInSignal* operation of the WDIManager interface.
17. The third-party application invokes the *destroyLSRSession* operation of the WDIManager interface.
18. The third-party application invokes the *destroyTransaction* operation on the WDIManager interface.
19. The third-party application invokes the *disconnect* operation of the WDIRoot interface.

LSR Field Validation Rules

General Rules

- If a *city* field is populated it must contain a city that exists in the MetaSolv Solution software's Infrastructure subsystem.
- If a *state code* field is populated it must contain a state code existing in the MetaSolv Solution Infrastructure subsystem.
- Telephone number fields must be numeric and be of appropriate length as specified by the OBF LSOG guidelines.

Importing LSOG 9 LSR Orders for NP

The LSR API assumes the LSR is LSOG compliant. Tables 8-16 define the validations that are performed by the MetaSolv Solution LSR API for importing LSOG 9 LSR (Local Service Request) orders for NP (Number Portability). These tables do not encompass all the rules the OBF guidelines suggest. However, these are additional validations that you might not infer from the OBF guidelines.

Table 8: LSR Admin. Field Validation Rules for Importing LSOG 9 Orders for NP

LSR Admin. Field	Rule
LSRADMIN.lsrno	This field is optional.
LSRADMIN.htqty	Required when reqeusting one or more hunt groups for this service request, otherwise prohibited.
LSRADMIN.an	Required when either the EAN field on the EU form or the ATN field is not populated or when a new AN is requested, otherwise optional.
LSRADMIN.atn	Required when either the EATN field on the EU Form or the AN field is not populated or when a new ATN is requested, otherwise optional.
LSRADMIN.sc1	Required if different from the SC field, otherwise optional.
LSRADMIN.resid	Prohibited when the RESID field on the LS Form is populated, otherwise optional.
LSRADMIN.dsptch	Optional when the ACT field is "N," "C," or "T," and the DSIND field on the preorder response is returned, otherwise prohibited.
LSRADMIN.dddo	1. Required when the service is to be suspended and the DDD field is populated with a restoral date. 2. Required for short term service (for example, trade shows) and the DDD field is populated with an install date. 3. Required for dual service or when the DDDO field is different from the DDD field for an outside move. 4. Other optional.
LSRADMIN.dfdt	Prohibited when the first position of the REQTYP field is "G", "H" or "J", otherwise optional.

Table 8: LSR Admin. Field Validation Rules for Importing LSOG 9 Orders for NP

LSR Admin. Field	Rule
LSRADMIN.p	Required when the first position of the REQTYP field is "N", otherwise prohibited.
LSRADMIN.sli	Optional when the first position of hte REQTYP field on the LSR Form is "A" or "B", other prohibited.
LSRADMIN.mi	Optional when the ACT field is "V" or "W", otherwise prohibited.
LSRADMIN.sup	1. Prohibited on initial requests. 2. Prohibited when changing a service inquiry to a firm order. 3. Prohibited when changing service type, which results in a change to the first position of the REQTYP field. 4. Otherwise optional.
LSRADMIN.exp	Required when the DDD field is less than the standard interval for the provisioning of the service, otherwise optional.
LSRADMIN.afo	Required when the associated request form(s) is applicable and sent, otherwise prohibited.
LSRADMIN.cc	Required when the CCNA field is "CUS", otherwise optional.
LSRADMIN.nnsp	Required when the first position of the REQTYP field is "B" or "C", the NPT field is "D" and the NNSP field entry is different than the entry in the CC field, otherwise optional.
LSRADMIN.onsp	Required when the first postion of the REQTYP field is "E", "F", "M", "N", or "P", and the NPI field is "C"or "D", otherwise optional.
LSRADMIN.agauth	Required when the customer is acting as an end user agent, otherwise optional.
LSRADMIN.dated	Required when the AGAUTH field is "Y", otherwise optional.

Table 8: LSR Admin. Field Validation Rules for Importing LSOG 9 Orders for NP

LSR Admin. Field	Rule
LSRADMIN.porttyp	Required when the first position of the REQTYP field is "F" or "M" and the second position of the TOS field is not "M" or "N" or the second position of the LTOS field on the PS Form is not "M" or "N", otherwise prohibited.
LSRADMIN.actl	1. Required when the first position of the REQTYP field is "A" or "B" and the ACT field is not "D" or "R". 2. Prohibited when the first position of the REQTYP field is "D", "E", "G", "H" or "J". 3. Otherwise optional.
LSRADMIN.sactl	Required when the SLI field is "A", otherwise prohibited.
LSRADMIN.ai	Required when the APOT field is populated, otherwise prohibited.
LSRADMIN.apot	Required when the ACTL field does not identify the specific physical termination point of service, otherwise optional.
LSRADMIN.lst	1. Required when the first position of the REQTYP field is "F" or "M" and the second postion of the TOS field is not "M" or "N" or the second position of the LTOS field on the PS Form is not "M" or "N". 2. Required when the first position of the REQTYP field is "E" and the entry is different than the end user's local serving office. 3. Otherwise optional.
LSRADMIN.tos	Required when the ACT field is "N", "C", "T", "V" or "W" and the first position of the REQTYP field is "E", "F" or "M" and the LTOS field on the service specific form is not populated, otherwise optional.
LSRADMIN.pbt	Optional when the ACT field is "N", "C", or "D" and the customer's ACTL field entry is a physical collocation arrangement, otherwise prohibited.
LSRADMIN.nci	Required when the NC field is populated, otherwise prohibited.
LSRADMIN.channel	Prohibited when the NC and NCI fields are populated, otherwise optional.

Table 8: LSR Admin. Field Validation Rules for Importing LSOG 9 Orders for NP

LSR Admin. Field	Rule
LSRADMIN.rord	Required when the provider has preassigned a related order number, otherwise prohibited.
LSRADMIN.lsp auth	1. Required when the second character of the TOS field is "P" or the second character of the LTOS field on the PS Form is "P" and the CCNA field is not equal to the ACNA field. 2. Required when the second character of the TOS field is "P" or the second character of the LTOS field on the PS Form is "P" and the CCNA field and the company represented in the CC field are not the same entity. 3. Otherwise optional.
LSRADMIN.lsp auth date	Required when the LSP AUTH field is populated, otherwise optional.
LSRADMIN.lsp auth name	Required when the LSP AUTH field is populated, otherwise optional.
LSRADMIN.npdi	Optional when the ACT field is "N", "C" or "V" and the first position of the REQTYP field is "C", otherwise optional.
EU.pon	Required. Must match LSR.pon.
EULocationAndAccess.locNum	Required. Must be sequential, consecutive numbers starting with 1.
EULocationAndAccess.Name	Required.
EULocationAndAccess.Sasn	Required.
EULocationAndAccess.City	Required.
EULocationAndAccess.State	Required.
EULocationAndAccess.ZipCode	Required.
EULocationAndAccess.ean	Required when the LSR.act = V and the EULocationAndAccess.eatn is not provided.
EULocationAndAccess.eatn	Required when the LSR.act = V and the EULocationAndAccess.ean is not provided.

Table 8: LSR Admin. Field Validation Rules for Importing LSOG 9 Orders for NP

LSR Admin. Field	Rule
NPServiceDetail.pon	Required. Must match LSR.pon.
NPServiceDetail.npQty	Required. Must be equal to the total number of ported telephone numbers for the entire order.
NPServiceDetail.locNum	Required. Must match an EULocationAndAccess.locNum on the order. Must be sequential, consecutive number.
NPServiceDetail.lNum	Required. Must be a sequential, consecutive number within the locNum. The combination of locNum/lNum must be unique at the pon level.
NPServiceDetail.lna	Required.
NPServiceDetail.portedNum (LSOG 3) NPServiceDetail.portedNbr (LSOG 4 and LSOG 5)	Required. Renamed from portedNum to portedNbr in LSOG 4. LSOG 3 supports a single telephone number. LSOG 4 and LSOG 5 support a single telephone number or a range of telephone numbers. All telephone numbers must exist in the MetaSolv Solution software's Telephone Number Inventory and be available to be ported out.
NPServiceDetail.npt	Required.
NPServiceDetail.cftn	Required when NPServiceDetail.npt = B, otherwise prohibited.
NPServiceDetail.nptg	Required when NPServiceDetail.npt = A, B, C and NPServiceDetail.lna = N or V.
NPServiceDetail.nptg	Required if NPServiceDetail.npt = A or C and NPServiceDetail.rti is not populated.
NPServiceDetail.rti	Required if NPServiceDetail.npt = A or C and NPServiceDetail.nptg is not populated.

Table 9: LSR Bill Section field Validation Rules for Importing LSOG 9 Orders for NP

Bill Section Field	Rule
LSRBILL.bi1	Required when the BAN1 and BAN2 fields are populated, otherwise optional.
LSRBILL.ban1	Required when a new billing account number is requested, otherwise optional.
LSRBILL.bi2	Required when the BAN2 field is populated, other optional.
LSRBILL.ban2	Required when the BI2 field is populated, otherwise prohibited.
LSRBILL.acna	Optional when the first position of the REQTYP field is "C" and the NPT field on the NP Form is "D", otherwise required.
LSRBILL.billnm	Required when the BAN (for example, BAN1 or BAN2) field is "N", otherwise optional.
LSRBILL.te	Required when the BAN (for example, BAN1 or BAN2) field is "N", otherwise optional.
LSRBILL.street	Required when the BAN (for example, BAN1 or BAN2) field is "N", otherwise optional.
LSRBILL.city	Required when the BAN (for example, BAN1 or BAN2) field is "N" and a billing profile is not established, otherwise optional.
LSRBILL.state	Required when the BAN (for example, BAN1 or BAN2) field is "N" and a billing profile is not established, otherwise optional.
LSRBILL.zip	Required when the BAN (for example, BAN1 or BAN2) field is "N" and a billing profile is not established, otherwise optional.

Table 9: LSR Bill Section field Validation Rules for Importing LSOG 9 Orders for NP

Bill Section Field	Rule
LSRBILL.billcon	Required when the BAN (for example, BAN1 or BAN2) field is "N", otherwise optional.
LSRBILL.tel no	Required when the BAN (for example, BAN1 or BAN2) field is "N", otherwise optional.

Table 10: LSR Contact Section field Validation Rules for Importing LSOG 9 Orders for NP

LSR Contact Section Field	Rule
LSRCONTACT.tel no(init)	Required field. No rule.
LSRCONTACT.tel no(dsg)	Required when the ALT IMPCON field is populated, otherwise prohibited.
LSRCONTACT.street	Required when the DSGCON field is populated, otherwise optional.
LSRCONTACT.city	Required when the DSGCON field is populated, otherwise optional.
LSRCONTACT.zip	Required when the DSGCON field is populated, otherwise optional.

Table 11: End User Admin. Section field Validation Rules for Importing LSOG 9 Orders for NP

EU Admin Section Field	Rule
EUADMIN.an	Required when either the EAN field or the ATN field is not populated or when a new AN is requested, otherwise optional.
EUADMIN.atn	Required when either the EATN field or the AN field is not populated or when a new ATN is requested, otherwise optional.
EUADMIN.adqty	Required when either the DISC NBR field or the DISC ECCKT field is populated, otherwise optional.

Table 12: Location and Access field Validation Rules for Importing LSOG 9 Orders for NP

Location and Access Field	Rule
Eua	Required when the MEU field on the LSR Form is populated, otherwise prohibited.
name	Optional when the SLI field on the LSR Form is "A", otherwise required.
Ncon	Optional when the ACT field on the LSR Form is "N", "T" or "V", otherwise prohibited.
Sano	Required when the AFT field is "C", otherwise optional.
Sasf	Optional when the SANO field is populated, otherwise prohibited.
Sasn	1. Optional when NPDI field on the LSR Form is populated with an "A" or "C". 2. Optional when the SLI field on the LSR Form is "A". 3. Otherwise required.
Lv1	Optional when the LD1 field is populated, otherwise prohibited.
Lv2	Optional when the LD2 field is populated, otherwise prohibited.
Lv3	Optional when the LD3 field is populated, otherwise prohibited.
City	1. Optional when NPDI field on the LSR Form is populated with and "A" or "C". 2. Optional when the SLI field on the LSR Form is "A". 3. Otherwise required.
State	1. Optional when NPDI field on the LSR Form is populated with an "A" or "C". 2. Optional when the SLI field onthe LSR form is "A". 2. Otherwise required.

Table 12: Location and Access field Validation Rules for Importing LSOG 9 Orders for NP

Location and Access Field	Rule
Zip1	1. Optional when NPDI field on the LSR Form is populated with an "A" or "C". 2. Optional when the SLI field onthe LSR form is "A". 2. Otherwise required.
Eumi	Required when the NPT field on either the LSNP Form or the NP Form is "D", otherwise optional.
Elt	Required when the ACT field on the LSR Form is "V", otherwise prohibited.

Table 13: Inside Wire field Validation Rules for Importing LSOG 9 Orders for NP

Inside Wire Field	Rule
iwcon	Required when the IWO field is populated, otherwise optional.
Tel no	Required when the IWCON field is populated, otherwise prohibited.

Table 14: Bill Section field Validation Rules for Importing LSOG 9 Orders for NP

Bill Section Field	Rule
Ean	Required when the ACT field on the LSR Form is "V" or "W" and the EATN field is not populated, otherwise optional.
Eatn	Required when the ACT field on the LSRForm is "V" or "W" and the EAN field is not populated, otherwise optional.
billnm	Required when the FBI field is "D", otherwise optional.
street	Required when the FBI field is "D", otherwise optional.

Table 14: Bill Section field Validation Rules for Importing LSOG 9 Orders for NP

Bill Section Field	Rule
City	Optional when NPDI field on the LSR Form is populated with an "A" or "C", otherwise required.
State	Optional when NPDI field on the LSR Form is poulated with an "A" or "C", otherwise required.
Zip	Optional when NPDI field on the LSR Form is poulated with an "A" or "C", otherwise required.
billcon	Required when the FBI field is populated and/or this entry is different from the BILLNM field, otherwise optional.
Telno	Required when the BILLCON field is populated, otherwise optional.

Table 15: Disconnect field Validation Rules for Importing LSOG 9 Orders for NP

Disconnect Field	Rule
Tc to pri	Required when the TC OPT field is not "blank" or "N", otherwise optional.
Tc to sec	Required when the TC OPT field is populated with a "T" or a customer code identifier that indicates split transfer of calls, otherwise prohibited.
Tcid	Required when the TC OPT field is "T" or a custom code identifier that indicates split transfer of calls, otherwise prohibited.
tcname	Required when the TC OPT field is "T" or a custom code identifier that indicates split transfer of calls, otherwise prohibited.
Tc per	Optional when the TC TO PRI field is populated, otherwise prohibited.

Table 16: NP field Validation Rules for Importing LSOG 9 Orders for NP

NP Field	Rule
NP.an	1. Required when the ATN field is not populated. 2. Required when the EAN field on the EU Form is "blank" or when a new AN is required. 3. Otherwise optional.
NP.atn	1. Required when the AN field is not populated. 2. Required when the EATN field on the EU Form is "blank" or when a new ATN is required. 3. Otherwise optional.
NP.tdt	Optional when the first position of the REQTYP field on the LSR Form is "C" and the NPT field is "D", otherwise prohibited.
NP.tnp	Required when the LNA field is "N", "T" or "V" and the NPT field is "A", "B" or "C", otherwise optional.
NP.cftn	Required when the NPT field is "B", otherwise optional.
NP.npt	Required when the LNA field is "N", "T" or "V", otherwise optional.
NP.lscp	1. Prohibited when the LSCP field is populated on the LSR Form. 2. Prohibited when the NPT field is "D". 3. Otherwise optional.
NP.block	1. Required when the BA field is "A" or "D". 2. Prohibited when the BA field is "blank". 3. Otherwise prohibited.

Table 16: NP field Validation Rules for Importing LSOG 9 Orders for NP

NP Field	Rule
NP.tc to pri	Required when the TC OPT field is "S", "T" or a custom code identifier, otherwise optional.
NP.tc to sec	Required when the TC OPT field is "T" or a custom code identifier that indicates split transfer of calls, otherwise prohibited.
NP.tcid	Required when the TC OPT field is "T" or a custom code identifier that indicates split transfer of calls, otherwise prohibited.
NP.tc name	Required when the TC OPT field is "T" or a custom code identifier that indicates split transfer of calls, otherwise prohibited.
NP.tc per	Optional when the TC TO PRI field is populated, otherwise prohibited.
NP.lean	Optional when the ACT field on the LSR Form is "V" and the EAN or EATN fields on the EU Form or the LEATN field is not populated, otherwise prohibited.
NP.leatn	Optional when the ACT field on the LSR Form is "V" and the EAN or EATN fields on the EU Form or the LEAN field is not populated, otherwise prohibited.