Table of Contents

1 Introduction
1.1 Overview ................................................................. 6
1.2 Purpose of this Guide .................................................. 6
1.3 XML Examples in this Document .................................... 6
1.4 Benefits of Exchange Shopping ..................................... 7

2 Understanding the Exchange Shopping Web Service
2.1 Introduction .................................................................. 9
2.2 Exchange Shopping Requirements ................................... 9
  2.2.1 Eligible Documents .................................................. 9
  2.2.2 Segments ............................................................... 9
  2.2.3 Passengers ............................................................ 10
  2.2.4 Shopping Options Returned ...................................... 10
2.3 Essential Terms .......................................................... 11
  2.3.1 Leg ........................................................................ 11
  2.3.1.1 EXS Xpath ........................................................ 11
  2.3.1.2 Example 1 – Single Segment Legs ......................... 11
  2.3.1.3 Example 2 – Multiple Segments Legs ................... 11
  2.3.2 Segment ............................................................... 13
  2.3.2.1 Exchange Shopping Xpath .................................... 13
  2.3.2.2 Example 1 ........................................................ 13
  2.3.2.3 Example 2 ........................................................ 13
  2.3.3 Leg Shopped vs. Leg Not Shopped ............................ 14
  2.3.3.1 EXS Xpath ........................................................ 15
  2.3.3.2 Example ............................................................ 15

3 Exchange Shopping Request XML
3.1 Introduction .................................................................. 16
3.2 Request Attributes ...................................................... 16
3.3 Request Elements ........................................................ 16
3.4 Chosen Elements of the Exchange Shopping Request ........ 17
  3.4.1 Point of Sale Information .......................................... 17
  3.4.1.1 Required Point of Sale Information ......................... 17
3.5 Passenger Information .................................................. 18
  3.5.1 Passenger with Active PNR ....................................... 18
  3.5.1.1 Required Information for Passenger with an Active PNR 18
  3.5.2 Passenger without Active PNR ................................... 19
  3.5.2.1 Required Information for a Passenger without an Active PNR 19
3.6 Travel Itinerary information ......................................... 20
  3.6.1 Three Types of Legs in Exchange Shopping Request ........ 20
  3.6.1.1 Leg that Is Not Shopped ....................................... 20
  3.6.1.1.1 Required Information for Leg that Is not Shopped .... 21
  3.6.1.2 Example of XML Request with Not Shopped Leg .... 21
  3.6.1.2 Leg that Is Shopped and “Use Same Connections” Shopping Parameter Is Applied .... 22
  3.6.1.3 Leg is Shopped and “Use Same Connections” Shopping Parameter is NOT Applied ........ 23
  3.6.1.3.1 Information on the Leg Level ............................ 23
  3.6.2 Removing Segments from Already Ticketed Itinerary......... 23
4 Exchange Shopping Searching and Pricing Parameters

4.1 Introduction ..........................................................25
4.2 Required Search Parameters ....................................25
  4.2.1 Origin and Destination City or Airport ....................25
  4.2.2 Departure or Arrival Date or Date and Time ..............26
4.3 Optional Searching and Pricing Parameters ..................27
  4.3.1 Search for Alternate Times before and/or after Departure Time ...........................................27
  4.3.2 Departure or Arrival Time Range ..........................27
  4.3.3 Search for Preferred Airline ...............................28
  4.3.4 Exclude an Airline ............................................29
  4.3.5 Specify Maximum Number of Connections Permitted per Leg .................................................29
  4.3.6 Specify Maximum Number of Stops Permitted per Leg .........................................................30
  4.3.7 Specify Online Carrier Service Only .......................30
  4.3.8 Specify Interline Carrier Service Only ..................31
  4.3.9 Consider Only These Connection Cities/Airports .........31
  4.3.10 Use Same Connections ......................................32
  4.3.11 Shop by Cabin Class .........................................32
    4.3.11.1 Jump Cabin Logic (default) .........................32
    4.3.11.2 Disable Jump Cabin Logic ...........................33
  4.3.12 Shop by Corporate ID .....................................33
  4.3.13 Shop by Account Code ....................................33
  4.3.14 Override the Passenger Type Code .......................34
    4.3.14.1 Requests for Infant PTCs Only .......................34
    4.3.14.2 Solution Returned in Different Booking Classes for Different PTCs ..............................35
    4.3.14.3 Solution Returned in Different Cabin Classes for Different PTCs ................................35
  4.3.15 Single Branded Fare ......................................36
  4.3.16 Government Status .........................................36
  4.3.17 Price no Penalty Fares ....................................36
  4.3.18 Consider Public/Private Fares Only .......................37
  4.3.19 Exempt all Taxes ...........................................37
  4.3.20 Price XO Fares ..............................................37

5 Exchange Shopping Response XML

5.1 Introduction ..........................................................38
  5.1.1 Response Attributes ........................................38
  5.1.2 Response Elements ...........................................38
5.2 Solutions Returned ..................................................39
  5.2.1 Solution Attributes ........................................39
  5.2.2 Solution Elements ...........................................42
    5.2.2.1 Information about Itinerary to be Booked ..........42
    5.2.2.2 Exchange Cost Breakdown per Each Passenger for Returned Itinerary .........................43
    5.2.2.3 Total Price Difference .................................48
6 Exchange Shopping Scenarios

6.1 Introduction .......................................................................................................................... 51
6.2 Fully Exchange – both legs shopped .................................................................................... 51
6.3 Partial Exchange – one leg already flown ........................................................................... 54
6.4 Exchange for multiple passengers ...................................................................................... 56
6.5 Non shopped leg ................................................................................................................... 61

7 Troubleshooting

7.1 Introduction .......................................................................................................................... 65
7.2 Error Messages by Category .............................................................................................. 65
7.2.1 Business Validation Errors ............................................................................................ 66
7.2.2.1 Table of Common Business Validation Errors ......................................................... 66
7.3 Errors from Other Subsystems ........................................................................................... 69
7.3.1 Unable to Verify Interline Coupon Status ...................................................................... 69
7.3.2 UNABLE TO REPRICE – VOLUNTARY CHANGE RULES FAILED ... 69
7.3.3 UNABLE TO REPRICE – VOLUNTARY CHANGE RULES UNAVAILABLE .................. 69
7.3.4 UNABLE TO MATCH FARES ....................................................................................... 69

8 Technical Support – Travel Agencies and 3rd party developers

8.1 Introduction .......................................................................................................................... 71
8.2 Phone .................................................................................................................................... 71
8.2 E-mail .................................................................................................................................... 71
Introduction

1.1 Overview

The Exchange Shopping web service (ExchangeShoppingRQv2.0.0) allows an agency customer, who needs to change a ticketed itinerary, to get information about available alternate itinerary options along with the exchange cost of each. With this solution, the customer will no longer need to go through a series of steps in order to find the most cost effective flight alternative for new desired travel dates and/or departure/arrival cities or airports.

The Exchange Shopping web service:
• Identifies fares in the ticket to be exchanged and, if all of them have Category 31 (Voluntary Changes) filed, returns available itinerary options that are Category 31 compliant and the exchange cost for each passenger in the request.
• Does not offer a solution to cancel the old itinerary and book the offered alternate solution after it comes to actual exchange.

1.2 Purpose of this Guide

This introductory guide provides:
• Description of web service features.
• Examples of typical exchange business scenarios.
• Examples of calls to Exchange Shopping.
• Common error messages that may be returned by Exchange Shopping along with the explanation as to why they occur and ways for fixing the problem where applicable.

1.3 XML Examples in this Document

The purpose of the XML examples provided in this document is to present the possible content of the request and response. The XML responses contain a limited number of solutions in order to save room, but at the same time provide you with a good sampling of responses.
1.4 Benefits of Exchange Shopping

Exchange Shopping allows agency customers to shop for available prices and routes when considering an itinerary change. Customers are able to consider various itinerary options and review the costs that would apply without having to alter their confirmed seats.

Exchange Shopping provides the following benefits:

- Allows shopping for alternate board points, off points and/or travel dates of a ticketed itinerary.
- Returns up to 50 itinerary solutions.
- Provides ability to initiate Exchange Shopping from unchanged PNRs or those that have been modified (segments added, deleted or changed).
- Provides ability to initiate Exchange Shopping when there is no PNR (Example: PNR was purged).
- Provides ability to add legs to the originally ticketed itinerary (Example: Changing it from a one-way to a round-trip fare).
- Provides ability to remove legs from the originally ticketed itinerary (Example: Changing it from round trip to one way).

In addition, customers are able to shop using the following optional qualifiers:

- Search for preferred airline.
- Exclude an airline or search with non-preferred airline.
- Specify maximum number of connections.
- Specify maximum number of stops.
- Specify online carrier service only.
- Specify interline carrier service.
- Search by date and time.
- Specify specific connection cities/airports.
- Use same connecting points.
- Shop by cabin class.
- Shop by corporate ID.
- Shop by Account Code.
- Override Passenger Type Code (PTC).
- Override default payment currency.
- Return single Brand option per itinerary.
- Price no penalty fares.
- Consider public/private fares only.
- Exempt all Taxes.
• Price XO fares.
Understanding the Exchange Shopping Web Service

2.1 Introduction

This chapter provides the description of Exchange Shopping. It provides information about types of transactions that the service supports and points out the ones that are not valid Exchange Shopping requests. It also contains an explanation of essential terms associated with the Exchange Shopping transactions used throughout this user guide.

2.2 Exchange Shopping Requirements

The following information provides the rules and applicability of Exchange Shopping.

2.2.1 Eligible Documents

- Exchange Shopping applies to flight documents only. Exchange Shopping does not apply to non-flight documents, such as electronic miscellaneous documents (EMDs).

- There must be only one flight document per passenger.

- Category 31 rules must be filed for all fares on the ticket being used to shop. If Category 31 is not available for any one of the fares on the ticket, an error message appears.

- Requests for multiple passengers must involve tickets issued for the exact same itinerary (marketing carrier(s) and flight number(s), booking class(es), board and off cities, travel date(s), coupon status(es)) and issued in the same currency.

  This applies to all un-flown, flown and previously flown coupons (if the ticket is a result of previous reissue after some coupons were already used).

2.2.2 Segments

Following are requirements pertaining to segments:

- Exchange Shopping applies for one way, round trip, open jaw, 2+ circle trip itineraries, and itineraries including up to a maximum of one side trip.

- Both online and interline itineraries are allowed.

- Each leg of travel must only involve a single travel date. Calendar shopping is not supported.

- When shopping for multiple passengers, all must have the same itinerary.

- The itinerary segments must be one of the following statuses: BK, HK, IK, RR, or SS.
OPEN segments are applicable to Exchange Shopping.

The new itinerary can be up to a maximum of 16 segments, including flown segments and surface sectors (ARNKs). If it is more than that, an error message appears.

### 2.2.3 Passengers

Following are requirements pertaining to passengers:

- You can shop for up to nine (9) passengers.
- The passengers can be different passenger types (PTCs). For example, 2ADT and 2CNN. There can be a maximum of four different types of PTCs.
- Exchange Shopping works with all currently processed passenger type codes (PTCs).
  Exception – The following standby passenger types do not apply to Exchange Shopping: ASB, BUD, AST, NSB, CSB, YSB, SDB, MSB, and YCB.
- A shopping request only for an infant (no seat) is not applicable to Exchange Shopping. You can include an infant with accompanying adults. For more information, see “Override Passenger Type Code,” page 35.

### 2.2.4 Shopping Options Returned

Following are requirements pertaining to shopping options returned:

- Exchange Shopping will return up to 50 options.
- The system can return shopping options in different booking classes or in different cabins. When this occurs, you will see flags advising of this in the XML.
- Exchange Shopping does not validate fares that require an accompanying passenger.
- The system checks seat availability for the total number of passengers traveling.
- A maximum of 16 segments is allowed.
- When checking options for infants (INF), Shopping may return alternate options where there is no seat available in the booking class for the given fare. Because the INF passenger type does not occupy a seat, Pricing does not verify seats availability to price. For more information, please see “Override Passenger Type Code,” page 35.
- Itineraries returned in an Exchange Shopping response will not include ancillary items (*Air Extras*).
2.3 Essential Terms

Following are terms related to Exchange Shopping that may help you understand the rules of forming requests to the service.

2.3.1 Leg

A Leg is a portion of travel in between two stopover places. A leg can be a single flight or consist of multiple connecting flights.

In this user guide, you may also see Leg referred to as portion of travel or O&D (Origin and Destination information).

The Pricing definition of stopover that applies to Exchange Shopping transactions is as follows:

- Countries except USA/Canada – A stopover is when a passenger arrives at an intermediate point and is scheduled to depart later than 24 hours after arrival (local time).
- USA/Canada – A stopover is when a passenger arrives at an intermediate point and is scheduled to depart later than 4 hours after arrival (local time).

2.3.1.1 EXS Xpath

ExchangeShoppingRQ/OriginDestinationInformation

2.3.1.2 Example 1 – Single Segment Legs

The following itinerary has two legs:

- First leg starting on 18 JAN at SVO and ending in MRV.
- Second leg starting on 23 JAN in MRV and ending at SVO.

```
1 U01304Y 18JAN J SVO MRV HK1 840A 1050A/E
2 U05371Y 23JAN Q MRV SVO HK1 640A 905A/E
```

2.3.1.3 Example 2 – Multiple Segments Legs

The following itinerary has two legs:

- First leg starting on 23 JAN at LED and ending at JFK
- Second leg starting on 29 JAN at JFK and ending at LED

```
1 U013Y 23JAN Q LED SVO HK1 1220P 135P/E
2 U0102Y 23JAN Q SVO JFK *HK1 340P 510P/E
3 U0101Y 29JAN W* JFK SVO HK1 210P 820A 30JAN Q/E
4 U012Y 30JAN Q SVD LED *HK1 1005A 1130A/E
```
Mandatory elements that define all legs are:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start location</td>
<td>Departure city/airport of this portion of travel is. Depending on whether</td>
<td>ExchangeShoppingRQ/OriginDestinationInformation/StartLocation</td>
</tr>
<tr>
<td></td>
<td>the leg is shopped, it can be three-letter city or airport code.</td>
<td></td>
</tr>
<tr>
<td>End location</td>
<td>Arrival city/airport of this portion of travel. Depending on whether the</td>
<td>ExchangeShoppingRQ/OriginDestinationInformation/EndLocation</td>
</tr>
<tr>
<td></td>
<td>leg is shopped, it can be three-letter city or airport code.</td>
<td></td>
</tr>
<tr>
<td>Departure or arrival</td>
<td>Date or date and time of either departure or arrival of given portion of</td>
<td>• ExchangeShoppingRQ/OriginDestinationInformation/DateTimeSelection/DepartureDate</td>
</tr>
<tr>
<td>date or date &amp; time</td>
<td>travel. In the case of legs that are not shopped, we recommend using</td>
<td>• ExchangeShoppingRQ/OriginDestinationInformation/DateTimeSelection/DepartureDateTime</td>
</tr>
<tr>
<td></td>
<td>departure date. In the case of legs that are shopped, use the date or date</td>
<td>• ExchangeShoppingRQ/OriginDestinationInformation/DateTimeSelection/ArrivalDate</td>
</tr>
<tr>
<td></td>
<td>and time search criteria, including additional optional time parameters if</td>
<td>• ExchangeShoppingRQ/OriginDestinationInformation/DateTimeSelection/ArrivalDateTime</td>
</tr>
<tr>
<td></td>
<td>required, that meet your needs the best.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For more information about available date or date and time search</td>
<td></td>
</tr>
<tr>
<td></td>
<td>parameters, see the following sections:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• “Departure or Arrival Date or Date and Time,” page 27.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• “Search for Alternate Times before and/or after Departure Time,” page</td>
<td></td>
</tr>
<tr>
<td></td>
<td>28.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• “Departure or Arrival Time Range,” page 28.</td>
<td></td>
</tr>
</tbody>
</table>

Notes

- Depending whether the leg is shopped or not and if shopped, what type of search parameters are used, also segment information with all necessary data may be required. For more information about segments, see the following section, as well as “Three Types of Legs in Exchange Shopping Request,” page 21.

- If there are multiple legs in the Exchange Shopping request, they should be sent in chronological order. If lack of chronology is detected by the business validation layer of the Exchange Shopping web service, an error message will be returned and the transaction will be terminated.
### 2.3.2 Segment

A *segment* is an air segment that is part of the leg.

### 2.3.2.1 Exchange Shopping Xpath

`ExchangeShoppingRQ/OriginDestinationInformation/RelatedSegment`

### 2.3.2.2 Example 1

There are two legs in the following itinerary, each consisting of one segment:

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>001304Y</td>
<td>18JAN</td>
<td>J</td>
<td>SVOMRV</td>
<td>HK1</td>
<td>840A</td>
</tr>
<tr>
<td>2</td>
<td>005371Y</td>
<td>23JAN</td>
<td>Q</td>
<td>MRVSVO</td>
<td>HK1</td>
<td>640A</td>
</tr>
</tbody>
</table>

- **Leg (OriginDestinationInformation) 1**: with start location at SVO, end location in MRV and departure date on 18 JAN
  - Segment (RelatedSegment) 1: U0 1304 Y 18JAN SVOMRV HK  840A 1050A

- **Leg (OriginDestinationInformation) 2**: with start location in MRV, end location in SVO and departure date on 23 JAN
  - Segment (RelatedSegment) 1: U0 5371 Y 23JAN MRVSVO HK  640A 905A

### 2.3.2.3 Example 2

There are two legs in following itinerary, each consisting of two segments (connecting flights):

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U0</td>
<td>13Y</td>
<td>23JAN</td>
<td>Q*</td>
<td>LEDsVO</td>
<td>HK1</td>
</tr>
<tr>
<td>2</td>
<td>U0</td>
<td>102Y</td>
<td>23JAN</td>
<td>Q</td>
<td>SVOJFK</td>
<td>*HK1</td>
</tr>
<tr>
<td>3</td>
<td>U0</td>
<td>101Y</td>
<td>29JAN</td>
<td>W*</td>
<td>JFKSVO</td>
<td>HK1</td>
</tr>
<tr>
<td>4</td>
<td>U0</td>
<td>12Y</td>
<td>30JAN</td>
<td>Q</td>
<td>SVOLED</td>
<td>*HK1</td>
</tr>
</tbody>
</table>

- **Leg (OriginDestinationInformation) 1**: with start location at LED, end location at JFK and departure date on 23 JAN
  - Segment (RelatedSegment) 1: U0 13Y 23JAN LEDSVO HK  1220P 135P
  - Segment (RelatedSegment) 2: U0 102Y 23JAN SVOJFK HK1  340P 510P

- **Leg (OriginDestinationInformation) 2**: with start location at JFK, end location at LED and departure date on 29 JAN
  - Segment (RelatedSegment) 1: U0 101Y 29JAN JFKSVO HK  210P 820A 30JAN
  - Segment (RelatedSegment) 2: U0 12Y 30JAN SVOLED HK  1005A 1130A
## 2.3.3 Leg Shopped vs. Leg Not Shopped

The terms “shopped” or “not shopped” leg are used to indicate whether (for a given portion of travel) the passenger wants to search for alternate itinerary options or leave this part of an already booked trip unchanged.

In the Exchange Shopping request XML, it is communicated by the user in the following attribute:

The following applies:

<table>
<thead>
<tr>
<th>When the given leg</th>
<th>And</th>
<th>Then set the shopIndicator value to</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is shopped</td>
<td>The Exchange Shopping web service should search for alternate itinerary solutions</td>
<td>“true”</td>
<td>At least one shopped leg per request is required to qualify for an Exchange Shopping transaction. If no leg was chosen to be shopped, an error message will be returned and the transaction terminated.</td>
</tr>
<tr>
<td>Is not shopped</td>
<td>The Exchange Shopping web service should not search for alternate options for this portion of travel, but keep its segments unchanged in all options returned to the customer</td>
<td>“false”</td>
<td>• If there are legs that are not shopped, options returned will include exactly the same flight segments within this leg (airline, flight number, date, etc.) except for the booking class which may be changed in order to secure best pricing solution. The exception is the scenario where flights that were marked as not shopped has the corresponding ticket coupons in checked-in (&quot;CKIN&quot;) or boarded (&quot;LFTD&quot;) status. In this case, when returning options Exchange Shopping will not propose rebooking to different class on these flights.</td>
</tr>
</tbody>
</table>
2.3.3.1 EXS Xpath

ExchangeShoppingRQ/OriginDestinationInformation/@shopIndicator

2.3.3.2 Example

Mr. Jones bought the ticket for the below itinerary (two legs):

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U01304Y</td>
<td>18JAN</td>
<td>J</td>
<td>SVOMRV HK1</td>
<td>0840</td>
<td>1050</td>
</tr>
<tr>
<td>2</td>
<td>U05371Y</td>
<td>23JAN</td>
<td>Q</td>
<td>MRVSVO HK1</td>
<td>0640</td>
<td>0905</td>
</tr>
</tbody>
</table>

His plans have changed and he found out that he needs to return from MRV 1 day earlier (22 JAN), however he wants to keep his outbound flight (SVO-MRV) unchanged.

In this scenario, the Exchange Shopping request needs to contain two legs:

- Leg (OriginDestinationInformation) 1: SVO-MRV departing on 18 JAN, shopIndicator="false"
  
  When given leg is not shopped, it will require providing full information about booked segment. For more information, see “Three Types of Legs in Exchange Shopping Request,” page 21).
  
  - Segment (RelatedSegment) 1: U01304Y 18JAN SVOMRV HK 840A 1050A

- Leg (OriginDestinationInformation) 2: MRV-SVO departing on 22 JAN, shopIndicator="true"
Exchange Shopping Request XML

3.1 Introduction

The purpose of this chapter is to briefly explain the content of the Exchange Shopping Request XML. The subsections of this chapter will provide detailed information about those parts of the XML request that may need some additional explanation over and above what is described in general information and schema annotations, and explain specifics of forming Exchange Shopping requests in different business scenarios.

Note This chapter should not be treated as complete information about the Exchange Shopping request XML schema, but can be used as additional source of information for the Exchange Shopping schema definition available in the DRC.

3.2 Request Attributes

Following are the request attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML version</td>
<td>Required in the XML schema.</td>
<td>ExchangeShoppingRQ/@Version</td>
</tr>
</tbody>
</table>

3.3 Request Elements

Following are the request elements:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>Standard header information that helps orchestrate the requests.</td>
<td>ExchangeShoppingRQ/Header</td>
</tr>
</tbody>
</table>

One of optional attributes of Header

| Default timeout override | Header element includes the attribute that allows the user to override the default timeout. When sent, it will be the time after which Exchange Shopping will try to stop processing the transaction. However if the system is in the middle of processing in one of the subsystems, the timeout may occur after the timeout sent by the user. The service will not be able to timeout in the middle of a process in the given subsystem. | ExchangeShoppingRQ/Header/@ttl     |
### Element Description

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remaining elements of Request XML</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Point of Sale (POS)</td>
<td>Information about the point of sale and the agent performing the Exchange Shopping transaction. This data must match the data that will be used in the subsequent exchange transaction; otherwise, the cost of the exchange may differ from the returned solution.</td>
<td>ExchangeShoppingRQ/POS</td>
</tr>
<tr>
<td>Ticketing provider code</td>
<td>Two-character code of the partition on whose behalf the Exchange Shopping transaction is performed. Agency customers should use the value “1S”</td>
<td>ExchangeShoppingRQ/TicketingProvider</td>
</tr>
<tr>
<td>Passenger information</td>
<td>Details about passengers and the flight documents that they want to exchange.</td>
<td>ExchangeShoppingRQ/PassengerInformation</td>
</tr>
<tr>
<td>Travel itinerary information</td>
<td>Information about the requested travel itinerary, including portions of travel (and related segments) that the passenger wants to leave unchanged if needed (legs not shopped) and/or those portions of travel for which the traveler wants to search for alternate options (legs shopped).</td>
<td>ExchangeShoppingRQ/OriginDestinationInformation</td>
</tr>
<tr>
<td>Travel preferences</td>
<td>Additional shopping criteria that should be used when searching for alternate options. All optional qualifiers, including the ones available under this part of the XML, are described in “Optional Searching and Pricing Parameters,” page 28.</td>
<td>ExchangeShoppingRQ/TravelPreferences</td>
</tr>
<tr>
<td>Diagnostic</td>
<td>For internal troubleshooting purposes only.</td>
<td>ExchangeShoppingRQ/Diagnostic</td>
</tr>
</tbody>
</table>

### 3.4 Chosen Elements of the Exchange Shopping Request

The subsections below provide more details about chosen elements of the Exchange Shopping request and some of their elements that may require some additional explanation.

**Note** This information should not be treated as complete information about the schema, but only as the supplement of XML schema documentation published in the DRC.

#### 3.4.1 Point of Sale Information

This section contains point of sale information.

#### 3.4.1.1 Required Point of Sale Information

Following is the list of required elements identifying the Point of Sale in an Exchange Shopping transaction. They are optional in the XML schema; however, their presence is verified by the service business validation layer.
### 3.5 Passenger Information

This part of the Exchange Shopping XML request was designed to collect information about passengers for whom the transaction is performed.

Depending on whether it is the scenario where an active PNR is available (for example, it was purged), a different set of data is required and described in the following sections.

#### 3.5.1 Passenger with Active PNR

This section contains information about passengers with an active PNR.

#### 3.5.1.1 Required Information for Passenger with an Active PNR

The following elements are required when the passenger has an active PNR. There are a maximum of nine passengers allowed per Exchange Shopping request; therefore, up to nine repetitions of this set of data are allowed.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Example</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>PNR locator</td>
<td>It is not required that all passengers in the request have the same PNR locator, however the requirement is exactly the same itinerary for all passengers in the request. If all passengers are not booked in the same PNR, it is the responsibility of the Point of Sale to handle the rebooking process correctly if the given alternate option is to be chosen.</td>
<td>YTRBDE</td>
<td>ExchangeShoppingRQ/PассенъrInformation/PassengerWithPNR/ @pnrLocator</td>
</tr>
</tbody>
</table>

---

Element | Description | Example | Exchange Shopping Path |
---|---|---|---|
Agent’s Pseudo City Code | Three-letter pseudo city code | G7HE | ExchangeShoppingRQ/POS/STL:Psuedo |
Shopping path | Allows the point of sale to specify the attribute that is used to determine shopping rules selection. The value of request type attribute passed by Point of Sale identifies the Shopping rule that should be applied during Exchange Shopping transaction. If there is no specific Shopping rule that the Point of Sale wants to apply, it is recommended to send the value “TNEXC”. | TNEXC | ExchangeShoppingRQ/POS/Shoppin Path/@requestType |
3.5.2 Passenger without Active PNR

This section contains information pertaining to passenger without an active PNR.

3.5.2.1 Required Information for a Passenger without an Active PNR

The following elements are required for the scenario where the passenger has no active PNR, for example, it was purged, but the ticket is still eligible for exchange. The assumption is that all portions of travel for passengers without an active PNR will be shopped.

There are a maximum of nine passengers allowed per Exchange Shopping request; therefore, up to nine occurrences of this set of data are allowed:
### Exchange Itinerary Information

This is the part of the XML request that provides information about the requested travel itinerary. The following sections provide tips on how requests should be formed in various business scenarios.

#### 3.6.1 Three Types of Legs in Exchange Shopping Request

Essentially, there are three types of legs (O&Ds) which, depending on whether they are shopped and whether the shopping search parameter Use Same Connections is applied, require a different set of data to be passed in the Exchange Shopping request.

**EXS Xpath:**

`ExchangeShoppingRQ/OriginDestinationInformation`

All three types of legs can be combined within a single Exchange Shopping request.

This section focuses on describing mandatory elements of the request depending on Exchange Shopping business scenarios and does not cover all optional search and pricing parameters available in this part of XML. For detailed information on these optional parameters, see “Optional Searching and Pricing Parameters,” page 28.

##### 3.6.1.1 Leg that Is Not Shopped

In the case of a leg that is not shopped, except elements describing the leg, full information about the segment that the passenger has already booked in the PNR and wants to keep in the new itinerary needs to be passed. The Exchange Shopping web service does not access the PNR to read this data but relies on information passed in Exchange Shopping request.

**EXS Xpath:**

`ExchangeShoppingRQ/OriginDestinationInformation/@shopIndicator="false"`
3.6.1.1.1 Required Information for Leg that Is not Shopped

This section provides information about elements that are required for flight segments being part of leg that is not shopped.

**Information on the Segment Level**

If the leg is not shopped, information about all related segments pertaining to this leg is required. If no flight segment information is provided for such a leg, an error message will be returned and transaction terminated.

The following flight segment information elements as appearing in PNR are required for the leg that is not shopped:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Example</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating carrier</td>
<td>Two-character alphanumeric code for the operating carrier of the flight segment.</td>
<td>U0</td>
<td>ExchangeShoppingRQ/OriginDest Information/RelatedSegment/@operatingProvider</td>
</tr>
<tr>
<td>Marketing carrier</td>
<td>Two-character alphanumeric code for the marketing carrier of the flight segment.</td>
<td>K0</td>
<td>ExchangeShoppingRQ/OriginDest Information/RelatedSegment/@marketingProvider</td>
</tr>
<tr>
<td>Flight number</td>
<td>Numeric marketing flight number.</td>
<td>001</td>
<td>ExchangeShoppingRQ/OriginDest Information/RelatedSegment/@marketingFlightNumber</td>
</tr>
<tr>
<td>Departure airport</td>
<td>Three-character departure airport of the flight segment.</td>
<td>JFK</td>
<td>ExchangeShoppingRQ/OriginDest Information/RelatedSegment/@startLocation</td>
</tr>
<tr>
<td>Arrival airport</td>
<td>Three-character arrival airport of the flight segment.</td>
<td>BOS</td>
<td>ExchangeShoppingRQ/OriginDest Information/RelatedSegment/@endLocation</td>
</tr>
<tr>
<td>Reservation booking class</td>
<td>Code for the reservation booking class.</td>
<td>Y</td>
<td>ExchangeShoppingRQ/OriginDest Information/RelatedSegment/@bookingClass</td>
</tr>
<tr>
<td>Reservation status</td>
<td>Reservation status as in PNR.</td>
<td>HK</td>
<td>ExchangeShoppingRQ/OriginDest Information/RelatedSegment/@reservationStatus</td>
</tr>
</tbody>
</table>

There are two additional attributes which are optional i.e …RelatedSegment/@startDate and …RelatedSegment/@endDate which are used to pass related segment departure and arrival date and time.

3.6.1.1.2 Example of XML Request with Not Shopped Leg

Scenario: The passenger would like to keep the first leg of his journey end exchange only the PHLLHR one

```
1.1JONES/MARK ADT 1 AA6130S 20MAY 6 LHRPHL HK1 1240 1525 /DCAA*NSSUOV /E
OPERATED BY BRITISH AIRWAYS
2 AA6129S 26MAY 5 PHLLHR HK1 1830 0635 27MAY 6
```
Following is an example of what the information about the leg that is not shopped would need to look like in the Exchange Shopping request:

**Exchange Shopping request XML**

```xml
<ExchangeShoppingRQ xmlns="http://services.sabre.com/sp/exchange/shopping/v2"
xmlns:n0="http://services.sabre.com/sp/exchange/shopping/common/types/v2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0.0">
  <STL_Header.RQ ttl="300">
    <OrchestrationID seq="0">02736434346758035</OrchestrationID>
  </STL_Header.RQ>
  <POS>
    <Pseudo>G7HE</Pseudo>
    <Actual>DFW</Actual>
    <ShoppingPath requestType="TNEXC" bookingChannel="TN"/>
  </POS>
  <TicketingProvider>1S</TicketingProvider>
  <PassengerInformation>
    <PassengerWithPNR pnrLocator="DNANZS" referenceNumber="1.1" firstName="MARK" lastName="JONES"/>
    <DocumentNumber>0017944161064</DocumentNumber>
  </PassengerInformation>
  <OriginDestinationInformation shopIndicator="false">
    <DateTimeSelection>
      <DepartureDate>2017-05-20</DepartureDate>
    </DateTimeSelection>
    <StartLocation>LHR</StartLocation>
    <EndLocation>PHL</EndLocation>
    <RelatedSegment operatingProvider="BA" marketingProvider="AA" marketingFlightNumber="6130" startDateTime="2017-05-20T12:40:00" endDateTime="2017-05-20T15:25:00" startLocation="LHR" endLocation="PHL" bookingClass="S" reservationStatus="HK" bookingDateTime="2017-05-10T04:08:00"/>
  </OriginDestinationInformation>
  <OriginDestinationInformation shopIndicator="true">
    <DateTimeSelection>
      <DepartureDate>2017-05-26</DepartureDate>
    </DateTimeSelection>
    <StartLocation>PHL</StartLocation>
    <EndLocation>BWI</EndLocation>
  </OriginDestinationInformation>
</ExchangeShoppingRQ>
```

### 3.6.1.2 Leg that is Shopped and “Use Same Connections” Shopping Parameter Is Applied

If the leg is shopped AND “Use Same Connections” shopping parameter are used for this leg, it means the passenger wants to search itinerary options for the indicated city pair and travel dates but wants the Exchange Shopping web service to return flight options for this leg that connect through the same airports as already booked. In such a case full related segment information is expected in the Exchange Shopping request for this leg.

**EXS Xpath:**

`ExchangeShoppingRQ/OriginDestinationInformation/@shopIndicator="true"`

and

`ExchangeShoppingRQ/OriginDestinationInformation/@preserveConnectionsIndicator="true"`
3.6.1.3 Leg is Shopped and “Use Same Connections” Shopping Parameter is NOT Applied

In the case of a leg that is shopped but the “Use Same Connections” search parameter is not applied:

EXS Xpath:

```
ExchangeShoppingRQ/OriginDestinationInformation/@preserveConnectionsIndicator="false" or is not sent at all
```

No related flight segment information is required. Even if sent, it will be ignored in processing. All necessary information is communicated on the leg level.

3.6.1.3.1 Information on the Leg Level

The following elements need to be passed for the leg that is not shopped:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airport or city code for travel start</td>
<td>Three-letter airport or city code of place where this portion of travel should start.</td>
<td><code>ExchangeShoppingRQ/OriginDestinationInformation/StartLocation</code></td>
</tr>
<tr>
<td>Airport or city code for travel end</td>
<td>Three-letter airport or city code of place where this portion of travel should end.</td>
<td><code>ExchangeShoppingRQ/OriginDestinationInformation/EndLocation</code></td>
</tr>
</tbody>
</table>
| Requested departure or arrival date or date and time when this portion of travel starts or ends | Departure or arrival date or date and time when this portion of travel should start or end. | • `ExchangeShoppingRQ/OriginDestinationInformation/DateTimeSelection/DepartureDate`
• `ExchangeShoppingRQ/OriginDestinationInformation/DateTimeSelection/DepartureDateTime`
• `ExchangeShoppingRQ/OriginDestinationInformation/DateTimeSelection/ArrivalDate`
• `ExchangeShoppingRQ/OriginDestinationInformation/DateTimeSelection/ArrivalDateTime` |

**Note**   
All other search and pricing parameters are optional. For detailed information about available options to search for alternate itineraries, see “Optional Searching and Pricing Parameters,” page 28.

3.6.2 Removing Segments from Already Ticketed Itinerary

If there are segments/legs that the user wants to delete from the booked itinerary, they should be omitted in the request sent to EXS-BSO. The Exchange Shopping request should contain only these segments that are going to be part of the new itinerary.
3.6.3 Maximum Number of Legs vs. Maximum Number of Segments

The maximum number of supported legs (O&Ds) is 10. They may still consist of maximum 16 segments.

EXS Xpath for leg:

*ExchangeShoppingRQ/OriginDestinationInformation*

EXS Xpath for segment:

*ExchangeShoppingRQ/OriginDestinationInformation/RelatedSegment*

The following sections present the logic of legs and segments count that is applied for Exchange Shopping request validation purposes.

3.6.3.1 Shopped Leg

If the leg is shopped, and the Use Same Connections search parameter is NOT applied, it is counted as one O&D and one segment at the same time.
4

Exchange Shopping Searching and Pricing Parameters

4.1 Introduction

The Exchange Shopping web service offers a wide range of searching and pricing parameters that passengers or airline agents can use to narrow or widen shopping results. Customers can use all of them or choose the ones that fit their business needs the best.

The purpose of this chapter is to provide detailed information about these parameters. Depending on the way they can be applied: per passenger, per leg, or per entire request, they are placed respectively in the following parts of the Exchange Shopping request XML:

EXS Xpath:

ExchangeShoppingRQ/PassengerInformation
ExchangeShoppingRQ/OriginDestinationInformation
ExchangeShoppingRQ/TravelPreferences

The following sections provide detailed information about each of them, specifying some business scenarios, describing expected results and referring to exact XML path where given parameters reside. The examples provided for each shopping parameter intend to present how the described qualifier can be used in the Exchange Shopping request XML, but obviously multiple searching qualifiers can be combined within a single request with exceptions described in this chapter.

4.2 Required Search Parameters

Exchange Shopping required parameters are to always be provided in the Exchange Shopping request in order to perform a search for alternate itinerary options.

4.2.1 Origin and Destination City or Airport

This qualifier provides the information about the origin and destination city/airport of given portion of travel that is applied during Exchange Shopping transaction. It needs to be provided if the user:

• Wants to leave the given portion of travel unchanged (leg not shopped).
• Wants to search for alternate options for a given leg (leg shopped), but wants to keep originally booked airports and change, for example, only the date.
• Wants to search for alternate options for a given leg changing board/off city(ies) or add a new leg to the already ticketed itinerary.
4.2.2 Departure or Arrival Date or Date and Time

This qualifier provides the information about the departure or arrival date or date and preferred time of the given portion of travel that is applied during the Exchange Shopping transaction. It needs to be provided when the user:

• Wants to leave a given portion of travel unchanged (leg not shopped).
• Wants to search for alternate options for a given leg (leg shopped), but wants to keep originally booked date and change, such as only departure city.
• Wants to search for alternate options for a given leg changing the travel date or add a new leg to the already ticketed itinerary.

The following applies:

• These parameters are applied per leg.
• Either departure OR arrival date or date and time can be requested per single leg.
• If both departure AND arrival date or date and time are requested per the same leg, an error message will be returned. This is controlled by the Exchange Shopping XML schema.
• If there is more than one leg within the request, departure or arrival dates or dates and times can be combined between different legs, for example, on the first leg, departure time is used and on the second one, arrival date and preferred time.
• The date/time is considered as local date/time at associated airport/city. For departure date/time, it is the local date/time at the origin airport/city, for arrival – local date/time at destination airport/city.
• If departure or arrival time is indicated in the request, the Exchange Shopping web service will search for the flights using the requested time as the preferred one.
• If no departure or arrival is sent in the request, the service will check for the flights departing or arriving between 00:00 and 23:59 of the day indicated as departure or arrival date. However, preferred flights will be the ones departing or arriving close to the time set as default time in DSF profiles.

EXS Xpaths:

ExchangeShoppingRQ/OriginDestinationInformation/DateTimeSelection/DepartureDate

or:

ExchangeShoppingRQ/OriginDestinationInformation/DateTimeSelection/DepartureDateTime

or:

ExchangeShoppingRQ/OriginDestinationInformation/DateTimeSelection/ArrivalDate

or:
4.3 Optional Searching and Pricing Parameters

Following are available optional searching and pricing parameters that can be applied by the user during the Exchange Shopping transaction to get expected results.

They are taken into consideration in the case of legs that are shopped. If set up for the leg that is not shopped, they are ignored in transaction processing.

4.3.1 Search for Alternate Times before and/or after Departure Time

This parameter allows searching for alternate itinerary options departing given number of hours before and/or after departure time specified in the request.

The following applies:

- Applicable per leg.
- The value can be from 001 to 009, which defines a window of 1 to 9 hours around the preferred departure time.
- “Alternate Times before and/or after Departure time” cannot be combined with Search by Arrival Date or Arrival Date and Time or Search by Departure/Arrival Time Range within the same leg.
- If the user shops for more than one leg, “Search for Alternate Times Before and/or After Departure Time” can be combined with Preferred Departure/Arrival Time Range, Search by Arrival Date or Arrival Date and Time between different legs.

EXS Xpath:

- \( ExchangeShoppingRQ/\text{OriginDestinationInformation}/\text{DateTimeSelection}/\text{DepartureDateTime}/@\text{minus} \) - departure given number of hours before departure time specified in the request
- \( ExchangeShoppingRQ/\text{OriginDestinationInformation}/\text{DateTimeSelection}/\text{DepartureDateTime}/@\text{plus} \) - departure given number of hours after departure time specified in the request

4.3.2 Departure or Arrival Time Range

This qualifier allows searching for itinerary solutions by departure or arrival time range.

The following applies:

- Applicable per leg.
- The default is to check all flight times.
- This applies to origin/destination only – not to intermediate flights.

Example

For the leg from Larnaca (LCA) to London (LON) via Amsterdam (AMS), the system considers the departure time range of flights from LCA or arrival time range in LON. AMS as connecting city is not considered.
• Either departure or arrival time range can be requested for a given leg.
• If departure/arrival time window is requested, both time window start and end must be passed; otherwise, the time range parameter will be ignored in processing.
• Combining departure or arrival time range with Preferred Departure or Arrival Time or “Alternate Time before/after Departure” searching parameters within a single leg is not allowed and controlled by the XML schema.

EXS Xpaths:

ExchangeShoppingRQ/OriginDestinationInformation/DateTimeSelection/DepartureDate/@timeWindowStart
ExchangeShoppingRQ/OriginDestinationInformation/DateTimeSelection/DepartureDate/@timeWindowEnd
or:
ExchangeShoppingRQ/OriginDestinationInformation/DateTimeSelection/ArrivalDate/@timeWindowStart
ExchangeShoppingRQ/OriginDestinationInformation/DateTimeSelection/ArrivalDate/@timeWindowEnd

4.3.3 Search for Preferred Airline

The Preferred Airline indicator narrows returned options only to the ones that consist of indicated carrier(s).

The following applies:
• Applicable per leg.
• The default is to check all carriers if carrier application is not restricted by Category 31 rules or other restrictions of the exchanged fare.
• Up to 20 preferred carriers per leg can be specified.
• The carrier code must be two alphanumeric characters. If any character other than alpha or numeric is entered or is not a two-character code, an error will be returned.
• This applies only to the marketing carrier. The operating carrier is not taken into consideration.
• If the preferred and non-preferred (see “Exclude an Airline,” page 30) carrier instructions are both included, the same carrier code should not be found in both. If the same airline is indicated, an error will be returned:

“Unable to process the request where the same airline: {carrier’s two-character code} is indicated as preferred and non-preferred carrier. Please amend your request and try again.”

Note  For this validation, only the legs that are shopped are taken into consideration.

EXS XPath:

ExchangeShoppingRQ/OriginDestinationInformation/IncludeVendor
4.3.4 Exclude an Airline

The Exclude an Airline indicator narrows returned options only to the ones that do not include non-preferred carrier(s).

The following applies:

- Applicable per entire request.
- The default is to check all carriers if the carrier application is not restricted by Category 31 rules or restrictions of any other category of exchanged fare.
- Up to 20 non-preferred carriers per itinerary can be specified.
- The carrier code must be two alphanumeric characters. If any character other than alpha or numeric is entered or these are not two characters, an error message will be returned.
- This applies only to the marketing carrier. The operating carrier is not taken into consideration.
- If the preferred (see “Search for Preferred Airline,” page 29) and non-preferred carrier instructions are both included, the same carrier code should not be found in both. If the same airline is indicated, an error will be returned:

  “Unable to process the request where the same airline: {carrier’s two-character code} is indicated as preferred and non-preferred carrier. Please amend your request and try again.”

EXS Xpath:

`exchangeShoppingRQ/travelPreferences/excludeVendor`

4.3.5 Specify Maximum Number of Connections Permitted per Leg

The Maximum Number of Connections indicator allows you to set the maximum acceptable number of connections allowed per leg in the itinerary. Intermediate stops with no change of plane are not counted as connections for the purposes of this indicator.

The following applies:

- Applicable per entire request (to all shopped legs). It is not possible to request a different maximum number of connections per each leg.
- Possible values are: 0, 1, 2, or 3 to indicate the maximum number of connections.
- The default is 3.
- If any other number is entered, an error message will be returned.
- Zero (0) indicates a request for nonstop or direct flights only.
- This qualifier cannot be combined with Use Same Connections and/or Specify Maximum Number of Stops per Leg.

EXS Xpath:

`exchangeShoppingRQ/travelPreferences/flight/@maxConnections`
4.3.6 Specify Maximum Number of Stops Permitted per Leg

This qualifier allows you to set the maximum acceptable number of stops allowed per portion of travel in the itinerary where intermediate stops with no change of plane are counted as connections for the purposes of this indicator.

The following applies:

- Applicable per entire request (to all shopped legs). It is not possible to request different number of maximum number of stops per each leg.
- Possible values are: 0, 1, 2, or 3 to indicate the maximum number of stops.
- If any other number than above entered, an error message will be returned.
- Zero (0) indicates a request for nonstop flights only.
- This qualifier cannot be combined with Use Same Connections and/or Specify Maximum Number of Connections per Leg.

EXS Xpath:

`ExchangeShoppingRQ/TravelPreferences/@maxStops`

4.3.7 Specify Online Carrier Service Only

This parameter allows searching for online (single carrier) itineraries.

The following applies:

- Applied per entire request.
- Each itinerary option will include flights of single (marketing) carrier only.
- Carriers may be different in different options (for example, itinerary option 1 can have flights for XX carrier, and itinerary option 2 can have flights for YY carrier only).
- The default is to search for both online and interline service itineraries if not restricted otherwise by Category 31 rules. If the user wants to request both online and interline itineraries, online and interline indicators (see “Specify Interline Carrier Service Only,” page 32) should be skipped in the request.
- If the online indicator (see the following EXS Xpath) is set to “true,” only itineraries on a single-carrier service are returned if applicable as per Category 31 rules of exchanged fare(s).
- This qualifier cannot be combined with “Specify interline carrier service only.” If both online and interline indicators are set to “true,” an error will be returned.
- This is applied only to the marketing carrier. Operating carrier is not taken into consideration.

EXS Xpath:

`ExchangeShoppingRQ/TravelPreferences/@onlineIndicator`
4.3.8 Specify Interline Carrier Service Only

This parameter allows searching for interline (multi-carrier service) itinerary options.

The following applies:

- Applied per entire request.
- If compliant with Category 31 restrictions of exchanged fare(s), each itinerary option will include flights of multiple carriers (at least two).
- The default is to search for both online and interline itineraries if not restricted otherwise by Category 31 rules. If the user wants to request both online and interline itineraries, online and interline indicators (see “Specify Online Carrier Service Only,” page 31) should be skipped in the request.
- If the interline indicator (see the following EXS Xpath is set to “true,” only itineraries with multiple-carrier service are returned if applicable as per Category 31 rules of exchanged fare(s).
- This qualifier cannot be combined with “Specify Interline Carrier Service Only.” If both online and interline indicators are set to “true,” an error will be returned.

EXS Xpath:

*ExchangeShoppingRQ/TravelPreferences/@interlineIndicator*

4.3.9 Consider Only These Connection Cities/Airports

This indicator allows specifying connection cities or airports via which the traveler wants to fly.

The following applies:

- When using city codes indicating connecting points, such as LON (London), Exchange Shopping will consider all airports belonging to specified city as per DSF settings for multi-airports city when searching for itinerary options.
- When using airport code as connecting point, such as LHR (London Heathrow airport), Exchange Shopping will consider only this airport when searching for the exchange itinerary options.
- When defining one connection city, Exchange Shopping will consider only this city/airport as connecting point when searching for exchange itinerary options.
- When defining more than one connection city/airport, Exchange Shopping will search for exchange itinerary options where one of the cities/airports is a connecting point or connecting points are constructed from the combination of indicated cities/airport. Options will not include flights via other cities/airports than indicated in the request.
- Direct flights can also be returned if connection locations are requested.
- It can be combined with Use Same Connections searching parameter but not on the same leg. If, for the same leg, both shopping parameters are used, an error message will be returned.

EXS Xpath:

*ExchangeShoppingRQ/OriginDestinationInformation/ConnectionLocation/Location*
4.3.10 Use Same Connections

This qualifier allows searching for exchange itinerary options using same connecting cities as already booked/ticketed by the traveler.

The following applies:

• This qualifier cannot be combined with Specify the Maximum Number of Stops per Leg and/or Specify the Maximum Number of Connections per Leg.

• It can be combined with Consider Only These Connection Cities/Airports, but not on the same leg. If, for the same leg, both shopping parameters are used, an error message will be returned.

• Exchange Shopping may return the options that have also some other connecting cities in addition to the indicated ones in the request if lower fares are found for such itineraries.

This qualifier cannot be used for transactions without an active PNR as this option requires full information about segments booked:

EXS Xpath:

ExchangeShoppingRQ/PassengerInformation/PassengerWithoutPNR

EXS Xpath:

ExchangeShoppingRQ/OriginDestinationInformation/@preserveConnectionsIndicator="true"

4.3.11 Shop by Cabin Class

This qualifier allows the customer specifying the cabin in which the itinerary options should be searched.

The following applies:

• Applicable per leg.

Jump Cabin Logic is enabled by default; however, depending on the value of Disable Jump Cabin element in the Exchange Shopping request:

EXS Xpath:

ExchangeShoppingRQ/TravelPreferences/@jumpCabinLogicDisabled

EXS Xpath to request specific cabin:

ExchangeShoppingRQ/OriginDestinationInformation/Cabin

4.3.11.1 Jump Cabin Logic (default)

The idea of Jump Cabin Logic is to search for solutions in lower or higher cabin if requested or defaulted cabin class is not offered or not available on searched itineraries.
4.3.11.2 Disable Jump Cabin Logic

The idea of Disable Jump Cabin Logic option (also known as *Expand Jump Cabin logic*) is to return only the flight options that have **at least one segment in the requested cabin**.

Disable Jump Cabin qualifier can be sent in the Exchange Shopping request or it can be switched on or off at the rule level in Intellisell Rules Manager.

**EXS Xpath:**

`ExchangeShoppingRQ/TravelPreferences/@jumpCabinLogicDisabled`

Options in rules settings in Intellisell Rules Manager referring to Disable Cabin Logic settings:

4.3.12 Shop by Corporate ID

This qualifier allows searching for negotiated fares using a Corporate ID.

The following applies:

- Applicable per entire request. If the request is for multiple passengers, such as two adults or one adult + one child, and the customer wants to use the Corporate ID to shop for exchange itineraries, this Corporate ID will be applied for the Exchange Shopping search for all passengers in the request.

- Exchange Shopping may return the lowest applicable fare combination, but it might not be the negotiated corporate rate. If the negotiated fare filed with Corporate ID is more expensive than cheapest public fare for this itinerary, it is the public fare that is going to be returned in Exchange Shopping.

- Only one Corporate ID per transaction is allowed.

- Corporate ID cannot be combined with Account Code within a single request. For more information about Account Code, see “Shop by Account Code,” page 34.

**EXS Xpath:**

`ExchangeShoppingRQ/TravelPreferences/PriceRequestInformation/@corporateID`

4.3.13 Shop by Account Code

This qualifier allows searching for negotiated fares using an Account Code.

The following applies:

- Applicable per entire request. If the request is for multiple passengers, such as two adults or one adult + one child, and the customer wants to use an Account Code to shop for exchange itineraries, this Account Code will be applied for the Exchange Shopping search for all passengers in the request.

- Negotiated fare with an Account Code requested is going to be returned as an exchange itinerary option only if there is no public fare available for the given itinerary that would be cheaper than the mentioned private fare.

- If the private fare with the Account Code is more expensive than the cheapest public fare for this itinerary, it is the published fare that is going to be returned by Exchange Shopping.
• Only one Account Code per transaction is allowed.

• Account Code cannot be combined with Corporate ID. For more information about Corporate ID see “Shop by Corporate ID,” page 34.

    EXS Xpath:

    ExchangeShoppingRQ/TravelPreferences/PriceRequestInformation/@accountCode

### 4.3.14 Override Passenger Type Code

This parameter allows searching for flight itinerary options using specific Passenger Type Code (PTC) for pricing purposes.

The following applies:

- Applicable per passenger and his/her whole itinerary.
- If for the given itinerary the adult (ADT) fare is cheaper than the fare for specified PTC and requested PTC is mapped to ADT, a cheaper ADT fare is going to be returned.
- Only one PTC can be specified per given passenger.
- When no PTC is defined in the Exchange Shopping Request, service will default the search to ADT PTC.
- Maximum of nine different PTCs are allowed per request.
- When checking options for infants (children under age 2 not occupying seats), such as INF, Exchange Shopping may return alternate options in the booking class that is not available. Because the infant type passenger type does not occupy a seat, service does not verify availability to price. This is in accordance with current pricing logic.

    EXS Xpath:

    ExchangeShoppingRQ/PassengerInformation/PassengerWithoutPNR/PassengerTypeOverride

or:

    ExchangeShoppingRQ/PassengerInformation/PassengerWithPNR/PassengerTypeOverride

#### 4.3.14.1 Requests for Infant PTCs Only

Exchange Shopping does not support requests for infant PTCs only.

This applies to PTCs overridden in Exchange Shopping request using the Shop by Passenger Type Code parameter or, if not applied, to the default PTC with which the exchanged ticket was priced.

*Example:*

Shopping for exchange options for:

- INF only, two INFs only or INF + JNF only,

will result in an error message returned by the service.

There is a requirement of at least one accompanying non-infant type passenger in the Exchange Shopping query in order to process transaction.
4.3.14.2 Solution Returned in Different Booking Classes for Different PTCs

When Exchange Shopping query involves multiple PTCs, such as two adults (ADT) and two youth passengers (YTH), the service might return an alternate itinerary solution in different booking classes for each PTC if this will secure best pricing solution.

This applies to PTCs overridden in the Exchange Shopping request using the Shop by Passenger Type Code parameter or, if not applied, to the defaulted PTC with which the exchanged ticket was priced.

**Note** The following example is for illustration purposes only and does not reflect exact content of the Exchange Shopping response XML.

*Example:*

Solution 1:

U0 811 20 OCT MEL SYD
U0 800 23 OCT SYD MEL

For ADTs, it may be returned in booking class V as this will be the best applicable pricing solution for this passenger type, and for passengers with YTH PTC in “Q” class as this class will offer the best price for this PTC.

When the solution includes different booking classes for different PTCs, the following flag will be set to “true” in order to indicate that fulfilling the exchange in offered fares will require splitting PNR:

**EXS Xpath:**

`ExchangeShoppingRS/Solution/@requireSplitPNR`

Still, the rule of checking seats availability for total number of passengers in the request is applied. In this scenario, it is four, so fares in V and Q class will be returned provided in each of these booking classes there are at least four seats available. This is designed to secure seats availability in “last seats” scenarios.

4.3.14.3 Solution Returned in Different Cabin Classes for Different PTCs

When the Exchange Shopping request involves multiple PTCs, such as two (adults) ADTs, two (youth) YTHs, Exchange Shopping might return an alternate itinerary solution that has not only different booking classes, but also cabins for each PTC if this will secure the best pricing solution, such as Business for ADTs and Economy for YTHs.

In the scenario, when an itinerary option returned in the Exchange Shopping response has a different cabin on at least one segment for different PTCs, the solution will contain a flag indicating this (along with the flag indicating necessity of PNR split described in the previous section):

**EXS Xpath:**

`ExchangeShoppingRS/Solution/@passengersInDifferentCabins="true"`

**Handling Infants in Terms of Options in Different Cabins**

If the itinerary solution includes a fare for infant type PTCs, such as INF, booked on at least one segment within this itinerary in a cabin that is different than for any other non-infant PTC(s), the option will be discarded from the Exchange Shopping response because the infant passenger cannot travel in a different cabin than the accompanying person.

**Note** There is no validation of the accompanying passenger for infant type PTCs performed.
4.3.15 Single Branded Fare

This qualifier will result in service adding brands information to each of the solutions.

If the brand IDs differ per fare component, service will mark the solution with parameter
ExchangeShoppingRS/Solution/Fare/mixedBrands

EXS Xpath:
ExchangeShoppingRQ/TravelPreferences/PriceRequestInformation/TPA_Extensions/BrandedFareIndicators/@singleBrandedFare

4.3.16 Government Status

This qualifier allows to price the fare that was filed with eligibility requirements. The condition is that a passenger must be a resident, employee or nationality of a given country to qualify for the fare. The type of the status (nationality, residency, employee) needs to be followed by appropriate country, country and state or city code.

- Nationality – requires 2-letter country code, e.g. RU (Russia),
- Residency – requires 2-letter country code, e.g. ES (Spain), or a 2-letter country code and 2-letter state/region code, e.g. US (USA) and FL (Florida); or requires a 3-letter city code, e.g. BKK (Bangkok),
- Employee requires 2-letter country code, e.g. ES (Spain) or requires 2-letter country code and 2-letter state/region code, e.g. US (USA) and FL (Florida).

EXS Xpath:
ExchangeShoppingRQ/TravelPreferences/PriceRequestInformation/GovernmentStatus

4.3.17 Price no Penalty Fares

This qualifier will look only for fares which have no change/refund fees.

EXS Xpath:
ExchangeShoppingRQ/TravelPreferences/PriceRequestInformation/@noPenaltyFares

4.3.19 Consider Public/Private Fares Only

These qualifiers will search only for published or private fares. They can be combined.

EXS Xpath:
ExchangeShoppingRQ/TravelPreferences/PriceRequestInformation/TPA_Extensions/PublicFare/@ind
ExchangeShoppingRQ/TravelPreferences/PriceRequestInformation/TPA_Extensions/PrivateFare/@ind
### 4.3.20  Exempt all taxes

This qualifier will exempt taxes only and include fees and passenger facility charges.

**EXS Xpath:**

```
ExchangeShoppingRQ/TravelPreferences/PriceRequestInformation/TPA_Extensions/ExemptAllTaxes/@value
```

### 4.3.21  Price XO fares

This qualifier will force fares with a single passenger type and ensure the response is for that passenger type only, even if there are lower fares.

**EXS Xpath:**

```
ExchangeShoppingRQ/TravelPreferences/PriceRequestInformation/TPA_Extensions/XOFares/@value
```
Exchange Shopping Response XML

5.1 Introduction

This section describes business specifics of returned itinerary options in the Exchange Shopping response.

The Exchange Shopping response (ExchangeShoppingRS) consists of the following attributes and elements.

5.1.1 Response Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML version</td>
<td>Required in the XML schema.</td>
<td>ExchangeShoppingRS/@Version</td>
</tr>
<tr>
<td>Number of exchange solutions returned</td>
<td>Required in the XML schema.</td>
<td>ExchangeShoppingRS/@solutions</td>
</tr>
</tbody>
</table>

5.1.2 Response Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solutions information</td>
<td>Detailed information about exchange itinerary options found based on customer's search criteria. Optional in the XML schema – it will be populated when at least one exchange itinerary solution is found.</td>
<td>ExchangeShoppingRS/Solution</td>
</tr>
</tbody>
</table>
## 5.2 Solutions Returned

**Note** This section is not intended to provide information about all elements and attributes in the Exchange Shopping response XML, but highlights only some of them that may appear under solution information.

### 5.2.1 Solution Attributes

Solutions returned may be marked with the following attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Example</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exchange Shopping response</td>
<td>Solutions are ordered from lowest to highest total price difference (price for all passengers).</td>
<td></td>
<td>ExchangeShopping</td>
</tr>
<tr>
<td>sequence number.</td>
<td></td>
<td></td>
<td>RS/Solution/@sequence</td>
</tr>
<tr>
<td>Pricing sequence</td>
<td>Pricing sequence number as received from one of the Exchange Shopping subsystems (IntelliSell). This attribute is used more for web service troubleshooting purposes.</td>
<td></td>
<td>ExchangeShopping.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RS/Solution/@pricingSequence</td>
</tr>
<tr>
<td>Required PNR split indicator</td>
<td>Indicator for whether the solution itinerary has different passenger types (PTC) booked in different booking classes, which would require a PNR split to fulfill the exchange transaction.</td>
<td>When the request contains different PTCs, such as two adults (ADT), two youth passengers (YTH), Exchange Shopping might return an alternate itinerary solution in different booking classes for each PTC if this will secure best pricing solution. Solution 1 (this example is only for illustration purposes and does not contain full XML content): U0 811 20 OCT MEL SYD U0 800 23 OCT SYD MEL For ADTs it may be returned in booking class “V” as this will be the best applicable pricing solution for this passenger type: U0 811  V 20 OCT MEL SYD U0 800  V 23 OCT SYD MEL And for passengers with YTH PTC in Q class as this class will offer best price for this PTC: U0 811 Q 20 OCT MEL SYD U0 800 Q 23 OCT SYD MEL When different booking classes are returned for a given itinerary solution, the following flag will be set to “true”:</td>
<td>ExchangeShopping.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RS/Solution/Fare/@requireSplitPNR</td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
<td>Example</td>
<td>Exchange Shopping Path</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>---------</td>
<td>------------------------</td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
<td>Example</td>
<td>Exchange Shopping Path</td>
</tr>
<tr>
<td>Same or different PTC evaluation logic:</td>
<td>Evaluation of same or different PTC is done based on the requested PTC. Therefore, in the scenario where there are two passengers, and they are shopped as ADT and YTH PTCs, but for given itinerary, the cheapest solution for both of them is for ADT passenger type, when returning options, Exchange Shopping may offer fares in different booking classes for these two passengers since they were requested as different PTCs.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Passengers booked in different cabins indicator</td>
<td>When the request contains different PTCs, such as two adults (ADT), two youth passengers (YTH), Exchange Shopping might return an itinerary solution that not only has different booking classes, but also cabins for each PTC if this will secure best pricing solution, such as Business for ADTs and Economy for YTHs.</td>
<td>ExchangeShopping RS/Solution/Fare/@passengersInDifferentCabin</td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
<td>Example</td>
<td>Exchange Shopping Path</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
<td>Example</td>
<td>Exchange Shopping Path</td>
</tr>
<tr>
<td>If the itinerary option returned from Shopping has infant type PTCs, such as INF, booked on at least one segment within this itinerary in a cabin that is different than for any other non-infant PTC(s), such as ADT, CNN, option will be discarded from the Exchange Shopping response because an infant passenger cannot travel in different cabin than accompanying person. There will be no validation of accompanying passenger data required by the fare rules for infant type PTCs performed.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Passengers priced in different currencies</td>
<td>Can be sent in the response for multiple passenger requests only. It indicates that exchange cost differences for passengers are calculated in different currencies and, as a result, Total Price Difference element summing the exchange cost for all passengers cannot be built. In such case, individual passenger’s prices should be used. This attribute is omitted in the response if exchange cost is calculated in the same currency for all passengers.</td>
<td></td>
<td>ExchangeShopping RS/Solution/Fare@passengersPricedInDifferentCurrencies</td>
</tr>
<tr>
<td>Valid</td>
<td>Indicates whether valid fare was found.</td>
<td></td>
<td>ExchangeShopping RS/Solution/Fare/valid</td>
</tr>
<tr>
<td>Brand</td>
<td>Adds the brand ID of the brand in which the price has been calculated</td>
<td></td>
<td>ExchangeShopping RS/Solution/Fare/brand</td>
</tr>
<tr>
<td>Branding Program</td>
<td>Adds branding program ID (up to 10 characters)</td>
<td></td>
<td>ExchangeShopping RS/Solution/Fare/brandingProgram</td>
</tr>
<tr>
<td>Mixed Brands</td>
<td>Indicates whether the itinerary was prices in mixed brands</td>
<td></td>
<td>ExchangeShopping RS/Solution/Fare/mixedBrands</td>
</tr>
</tbody>
</table>
5.2.2 Solution Elements

There are three main elements that the solution consists of:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information about the itinerary to be booked</td>
<td>Information about the itinerary to be booked.</td>
<td>ExchangeShoppingRS/Solution/BookItinerary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExchangeShoppingRS/Solution/Fare/ReservationSegmentDetails</td>
</tr>
<tr>
<td>Exchange price difference information per passenger</td>
<td>Exchange price difference information per passenger.</td>
<td>ExchangeShoppingRS/Solution/Fare/PassengerPriceInformation</td>
</tr>
<tr>
<td>Total exchange price difference information</td>
<td>Total exchange price difference information.</td>
<td>ExchangeShoppingRS/Solution/Fare/TotalPriceDifference</td>
</tr>
</tbody>
</table>

In the following sections, you will find the description of some of elements that you may see in each of them that may help you understand the content and business scenarios when they may appear with given values.

5.2.2.1 Information about Itinerary to be Booked

Itinerary information is split into legs. They are returned in the same order and number as requested in the Exchange Shopping request XML.

**Note** If the new itinerary contains a surface sector (ARNK) between the portions of travel, such as first leg ends in New York JFK airport (JFK) and second starts in Boston (BOS), information about the ARNK segment will be omitted in the Exchange Shopping response.

If there are any flown ticket coupons (scenarios with partially used tickets) that are not part of new itinerary calculation (travel itinerary information part of Exchange Shopping request XML), they will be taken into consideration for re-pricing purposes, but will not be added to the Exchange Shopping response.

**EXS Xpath:**

- ExchangeShoppingRS/Solution/BookItinerary/OriginDestination
- ExchangeShoppingRS/Solution/Fare/ReservationSegmentDetails
Under leg data, the following information appears:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information about each flight segment pertaining to the leg</td>
<td>The reservation segments that must be booked, priced, and fulfilled if the passenger decides to exchange his/her ticket for this itinerary. If the returned solution contains a surface sector, such as due to the airport change (example: from London Heathrow (LHR) to London Gatwick (LGW)), it will be omitted in the segment information.</td>
<td>ExchangeShoppingRS/Solution/BookItinerary/OriginDestination/ReservationSegment</td>
</tr>
<tr>
<td>Information about flight segment referred to each passenger separately</td>
<td>Place where the booking class and fare basis applicable for a given passenger is presented. All passengers are listed in same order as in the PassengerPriceInformation section of the XML response. The document number is used to uniquely identify the passenger and his/her booking and price information.</td>
<td>ExchangeShoppingRS/Solution/Fare/ReservationSegmentDetails/PassengerBookingDetails</td>
</tr>
</tbody>
</table>

**Note**  As mentioned earlier in this section, if there are some flown coupons in the shopped ticket(s), they will be considered during the re-pricing processing, but information about them (such as flight details and fare) will not be included in the Exchange Shopping response.

### 5.2.2.2 Exchange Cost Breakdown per Each Passenger for Returned Itinerary

Passengers (up to nine) may be in a different order than listed in the Exchange Shopping request.

Each passenger is identified by following information:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticket number</td>
<td>Ticket number.</td>
<td>ExchangeShoppingRS/Solution/FarePassengerPriceInformation/Passenger/@documentNumber</td>
</tr>
<tr>
<td>Passenger Type Code</td>
<td>Code with which the itinerary was eventually priced.</td>
<td>ExchangeShoppingRS/Solution/FarePassengerPriceInformation/Passenger/@type</td>
</tr>
</tbody>
</table>

*Example:* The passenger requested Exchange Shopping to return alternate solutions using the “Override Passenger Type Code” pricing qualifier and set it to YTH. However, for a given
itinerary, the system identified either of the following:

- No fares are available for YTH.
- Lowest applicable fare is for regular adult (ADT) passenger while YTH fare would be more expensive, or would not meet the exchanged ticket fare conditions.

and returned a fare for ADT. In this case, ADT would be populated as passenger type in the response for this itinerary.

Passenger’s first and last name

The following applies:
- For requests with active PNR, the name is the same as passed in Exchange Shopping request.
- For requests without active PNR, the name passed in the response is taken from exchanged ticket.

Age (optional)

Exchange price difference information per passenger

Detailed information about the cost of exchange to a given itinerary per each passenger.

Some of the attributes that are (or can be) added, depending on the scenario, are:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exchange result</td>
<td>Where possible, options are:</td>
<td>ExchangeShoppingRS/Solution/Fare/PassengerPriceInformation/Passenger/ResultPriceDifference@differenceType</td>
</tr>
<tr>
<td></td>
<td>• Additional collection</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(AddCollect) – indicates the passenger needs to pay some additional amount to fulfill this itinerary solution.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Even exchange (Even)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>indicates there is no additional cost to fulfill this itinerary solution.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Refund – indicates there is a refund due to the passenger that will be returned when fulfilling this itinerary solution.</td>
<td></td>
</tr>
</tbody>
</table>
Note  In the scenario where there is a payment currency change, currency in which the exchange price difference is returned may be different depending on result type:

- ADDITIONAL COLLECTION – default currency of country where Exchange Shopping transaction is performed or, if the customer used Payment Currency Code Override pricing parameter, in override (requested) currency.

- EVEN EXCHANGE or REFUND

<table>
<thead>
<tr>
<th>For</th>
<th>The exchange price difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>The first payment currency change</td>
<td>Will be returned in the currency in which the exchanged ticket was paid.</td>
</tr>
<tr>
<td>Subsequent exchanges with multiple payment currency changes</td>
<td>Will be returned in the default currency of the country where the Exchange Shopping transaction is performed or, if the customer used the Payment Currency Code Override pricing parameter, in the override (requested) currency.</td>
</tr>
</tbody>
</table>

Following are examples that illustrate these two scenarios.

**Example 1 – single payment currency change**

1. Original ticket issued on route: JFK-MOW-LON:
   - Base currency: USD
   - Issue place: LON, therefore it is paid in GBP (equivalent currency)

2. When shopping for exchange options for this ticket, the passenger is in MOW (default reissue currency RUB) or he is still in LON, but uses the Payment Currency Code Override pricing parameter to get exchange cost in RUB.

   Therefore, in this scenario:
   - Base currency: unchanged – USD
   - Reissue currency: RUB

Similarly to how Automated Exchange and Refunds (AER) works today, depending on the exchange result type, the exchange price difference is calculated in different currencies:

<table>
<thead>
<tr>
<th>For</th>
<th>The passenger gets exchange cost amounts in</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDITIONAL COLLECTION</td>
<td>RUB (default or override reissue currency).</td>
</tr>
<tr>
<td>REFUND or EVEN EXCHANGE</td>
<td>GBP (currency in which exchanged ticket was paid (equivalent currency in this case)).</td>
</tr>
</tbody>
</table>
Note  For scenarios where all three types of results are returned

Example 2 – Multiple payment currency change over subsequent exchanges

1. Original ticket issued on route: JFK-MOW-LON:
   - Base currency: USD
   - Issue place: LON; therefore, the ticket was paid in GBP (equivalent)

2. Ticket reissued in ZRH:
   - Base currency: USD
   - Reissue place: ZRH; therefore, reissue (equivalent) currency was CHF.

3. When shopping for exchange options for this ticket, the passenger is in MOW (default reissue currency RUB) or he is still in LON, but uses the Payment Currency Code Override pricing parameter to get exchange cost in RUB.
   - Therefore in this scenario:
     - Base currency: Unchanged – USD
     - Current reissue currency: RUB

Similarly to how Automated Exchange and Refunds (AER) works today, regardless of result of exchange (ADDITIONAL COLLECTION, REFUND or EVEN EXCHANGE), the passenger gets the exchange cost amounts in RUB (default or override reissue currency).
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Residual amount forfeited flag</td>
<td>Returned for these solutions where new fare is cheaper than the old fare, but the residual amount was forfeited per Category 31 rules of the exchanged fare and is not included in the grand total difference amount. If not applicable for given itinerary, then it is omitted in the response.</td>
<td>ExchShoppingRS/Solution/Fare/Passenger/PriceInformation/Passenger/ResultPriceDifference/@ResidualAmountForfeited</td>
</tr>
</tbody>
</table>

**Exchange price difference elements (elements that are part of exchange cost breakdown)**

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount and currency of difference in fare</td>
<td>Amount and currency of difference in fare between exchanged ticket fare and new itinerary fare.</td>
<td>• ExchShoppingRS/Solution/Fare/Passenger/PriceInformation/Passenger/ResultPriceDifference/FareDifference/@currencyCode&lt;br&gt;• ExchShoppingRS/Solution/Fare/Passenger/PriceInformation/Passenger/ResultPriceDifference/FareDifference/@decimalPlace</td>
</tr>
<tr>
<td>Amount and currency of difference in taxes</td>
<td>Amount and currency of difference in taxes (sum) between exchanged ticket and new itinerary.</td>
<td>• ExchShoppingRS/Solution/Fare/Passenger/PriceInformation/Passenger/ResultPriceDifference/TaxDifference/@currencyCode&lt;br&gt;• ExchShoppingRS/Solution/Fare/Passenger/PriceInformation/Passenger/ResultPriceDifference/TaxDifference/@decimalPlace</td>
</tr>
<tr>
<td>Amount and currency of subtotal difference</td>
<td>Amount and currency of subtotal difference (sum of fare difference and taxes difference).</td>
<td>• ExchShoppingRS/Solution/Fare/Passenger/PriceInformation/Passenger/ResultPriceDifference/SubtotalDifference/@currencyCode&lt;br&gt;• ExchShoppingRS/Solution/Fare/Passenger/PriceInformation/Passenger/ResultPriceDifference/SubtotalDifference/@decimalPlace</td>
</tr>
<tr>
<td>Total amount, currency and type of change fee</td>
<td>Total amount, currency and type of change fee applied to exchange. Fee total is split by fee type. Supported types are: CHG – Change fees, OTH – Other fees</td>
<td>• ExchShoppingRS/Solution/Fare/Passenger/PriceInformation/Passenger/ResultPriceDifference/TotalFee/@currencyCode&lt;br&gt;• ExchShoppingRS/Solution/Fare/Passenger/PriceInformation/Passenger/ResultPriceDifference/TotalFee/@decimalPlace&lt;br&gt;• ExchShoppingRS/Solution/Fare/Passenger/PriceInformation/Passenger/ResultPriceDifference/TotalFee/@type</td>
</tr>
<tr>
<td>Element</td>
<td>Description</td>
<td>Exchange Shopping Path</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>------------------------</td>
</tr>
</tbody>
</table>
| Total amount and currency of tax on change fee | If not applied, it is omitted in the response. | • ExcShoppingRS/Solution/Fare PassengerPriceInformation/Passenger/ResultPriceDifference/TotalFeeTax/@currencyCode  
• ExcShoppingRS/Solution/Fare PassengerPriceInformation/Passenger/ResultPriceDifference/TotalFeeTax/@decimalPlace |

| Amount and currency of grand total difference | If the exchange results in additional collection for the passenger, this is a positive amount  
If the exchange results in a refund due to the passenger and zero amount if this is an even exchange, this is a negative amount. | • ExcShoppingRS/Solution/Fare PassengerPriceInformation/Passenger/ResultPriceDifference/GrandTotalDifference/@currencyCode  
• ExcShoppingRS/Solution/Fare PassengerPriceInformation/Passenger/ResultPriceDifference/GrandTotalDifference/@decimalPlace |

### 5.2.2.3 Total Price Difference

This part of the Exchange Shopping XML response provides information about the total cost of the exchange (total amount to be paid or refunded to the customer) for all passengers, so the customer knows the final total exchange cost for a given itinerary solution.

One of the attributes of the total price difference is to provide the difference result again, but this time, it is calculated based on exchange costs for all passengers.

It can have the following values:

- **Additional collection** (AddCollect) – indicates that passenger needs to pay some additional amount to fulfill this itinerary solution for all passengers in the request.
- **Even exchange** (Even) – indicates that there is no additional cost to fulfill this itinerary solution.
- **Refund** – indicates that there is a refund due to the passenger that will be returned when fulfilling this itinerary solution for all passengers in proposed fares.

**EXS Xpath:**

*ExchangeShoppingRS/Solution/Fare/TotalPriceDifference/@differenceType*

**Note**  
Total Price Difference element is returned in itinerary solutions where the exchange cost difference is calculated in the same currency for all passengers. In scenarios of multiple passenger requests where passengers are priced in different currencies, this element is omitted in the response and the flag is set to “true” for such itinerary solution:

**EXS Xpath:**

*ExchangeShoppingRS/Solution/Fare/@passengersPricedInDifferentCurrencies*
This part of the XML response contains the following Total price difference elements representing sums for all passengers:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Exchange Shopping Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount and currency of difference in fare</td>
<td>Amount and currency of difference in fare between exchanged tickets fare and new itinerary fares.</td>
<td>• ExcShoppingRS/Solution/Fare TotalPriceDifference/FareDifference/@currencyCode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ExcShoppingRS/Solution/Fare TotalPriceDifference/FareDifference/@decimalPlace</td>
</tr>
<tr>
<td>Amount and currency of difference in taxes</td>
<td>Amount and currency of difference in taxes between exchanged tickets and taxes on new itinerary.</td>
<td>• ExcShoppingRS/Solution/Fare TotalPriceDifference/TaxDifference/@currencyCode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ExcShoppingRS/Solution/Fare TotalPriceDifference/TaxDifference/@decimalPlace</td>
</tr>
<tr>
<td>Amount and currency of subtotal difference</td>
<td>Amount and currency of subtotal difference (sum of fare differences and taxes differences).</td>
<td>• ExcShoppingRS/Solution/Fare TotalPriceDifference/SubtotalDifference/@currencyCode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ExcShoppingRS/Solution/Fare TotalPriceDifference/SubtotalDifference/@decimalPlace</td>
</tr>
<tr>
<td>Total amount, currency and type of change fees</td>
<td>Total amount, currency, and type of change fees applied to exchange for all passengers. Fees totals are split by fee type. Supported types are:</td>
<td>• ExcShoppingRS/Solution/Fare TotalPriceDifference/TotalFee/@currencyCode</td>
</tr>
<tr>
<td></td>
<td>• CHG - Change fees</td>
<td>• ExcShoppingRS/Solution/Fare TotalPriceDifference/TotalFee/@decimalPlace</td>
</tr>
<tr>
<td></td>
<td>• OTH - Other fees</td>
<td>• ExcShoppingRS/Solution/Fare TotalPriceDifference/TotalFee/@type</td>
</tr>
<tr>
<td>Total amount and currency of taxes on change fees</td>
<td>Total amount and currency of taxes on change fees if applicable. If not applied, it is omitted in the response.</td>
<td>• ExcShoppingRS/Solution/Fare TotalPriceDifference/TotalFeeTax/@currencyCode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ExcShoppingRS/Solution/Fare TotalPriceDifference/TotalFeeTax/@decimalPlace</td>
</tr>
<tr>
<td>Element</td>
<td>Description</td>
<td>Exchange Shopping Path</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Amount and currency of grand total difference</td>
<td>Amount and currency of grand total difference.</td>
<td>• <strong>ExcShoppingRS/Solution/Fare</strong> TotalPriceDifference/GrandTotalDifference/@currencyCode</td>
</tr>
<tr>
<td></td>
<td>If the exchange for all passengers results in an additional collection, it is a positive amount.</td>
<td>• <strong>ExcShoppingRS/Solution/Fare</strong> TotalPriceDifference/GrandTotalDifference/@decimalPlace</td>
</tr>
<tr>
<td></td>
<td>If the exchange for all passengers results in a refund due to the customer, it is a negative amount.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the exchange results in an even transaction (no additional collection or refund due), it is a zero amount.</td>
<td></td>
</tr>
</tbody>
</table>
Exchange Shopping Scenarios

6.1 Introduction

The following examples provide sample XML requests and responses for more complex Exchange Shopping transactions including various scenarios presenting the ways in which the web service can be used.

6.2 Full Exchange – both legs shopped

```
*SBXGVL*
1.1PETRICEK/WITOLD 2.1CUTO/FLORENCIA
1 AA 8JY 20MAY J LH89DFW HK2 1250P 505P /DCAA*SBXGVL /E
2 AA 5DY 16JUN J DFWLHR HK2 350P 655A 11JUN 8
                     /DCAA*SBXGVL /E

TKT/TIME LIMIT
1.T-11APR-17HE*AST
PHONES
  1.ABC123456
INVOICED
PRICE QUOTE RECORD EXISTS - SYSTEM
SECURITY INFO EXISTS *P3D OR *P4D TO DISPLAY
REMARKS
  1.XXTAN/
ACCOUNTING DATA
  1. AA794161067/ 0.00/ 2398.00/ 485.96/ONE CA 1.1STRI
     C28K WITOLD/1/E/B
  2. AA794161068/ 0.00/ 2398.00/ 485.96/ONE CA 2.1CUTO
     FLORENCIA/1/E/S
RECEIVED FROM - SF TEST
C7HE.C7HE*AST 1010/11APR17 SBXGVL H

ELECTRONIC TICKET RECORD
INV: 0017994161068 CUST: PNR: SBXGVL
TKT: 0017994161068 ISSUED: 11APR17 PCC: C7HE IATA: 999999999
NAME: CUTO/FLORENCIA
EOP: CHECK
ODN A/L PLT CLS DATE BRDOFF TIME ST E/B STAT
  1 AA 81 Y 20MAY LH89DFW 1250P CK X1NU4C5 ODEN
  2 AA 50 Y 10JUN DFWLHR 350P CK X1NU4C5 OPEN
FARE GBP1937.00 TAX 283.40YR TAX 36.00US TAX 5.50YC
  TAX 7.00XK TAX 3.95YA TAX 5.60AY
  TAX 87.95GB TAX 52.10UB TAX 4.50XP
TOTAL USD2883.96 EQUIV FARE ED USD2398.00
LON AA DFWL181.51AA LCN181.51NUC2363.02END ROE0.819712 XFD6W4.5
```
Both passengers wish to change both their destinations and dates of travel:

```xml
<ExchangeShoppingRQ xmlns="http://services.sabre.com/sp/exchange/shopping/v2"
xmlns:n0="http://services.sabre.com/sp/exchange/shopping/common/types/v2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0.0">
  <STL_Header.RQ n0:seq="0" n0:OrchestrationID="027364343467758035"></STL_Header.RQ>
  <POS>
    <Pseudo>G7HE</Pseudo>
    <Actual>DFW</Actual>
    <ShoppingPath requestType="TNEXC" bookingChannel="TN"/>
  </POS>
  <TicketingProvider>1S</TicketingProvider>
  <PassengerInformation>
    <PassengerWithPNR pnrLocator="SBXGYL" referenceNumber="1.1" n0:firstName="WITOLD" n0:lastName="PETRICZEK">
      <DocumentNumber>0017944161067</DocumentNumber>
    </PassengerWithPNR>
    <PassengerWithPNR pnrLocator="SBXGYL" referenceNumber="2.1" n0:firstName="FLORENCIA" n0:lastName="COUTO">
      <DocumentNumber>0017944161068</DocumentNumber>
    </PassengerWithPNR>
  </PassengerInformation>
  <OriginDestinationInformation shopIndicator="true">
    <DateTimeSelection>
      <DepartureDate>2017-05-20</DepartureDate>
      <DateTimeSelection>
        <StartLocation>LON</StartLocation>
        <EndLocation>NYC</EndLocation>
      </DateTimeSelection>
      <OriginDestinationInformation>
        <OriginDestinationInformation shopIndicator="true">
          <DateTimeSelection>
            <DepartureDate>2017-06-10</DepartureDate>
            <DateTimeSelection>
              <StartLocation>LON</StartLocation>
              <EndLocation>NYC</EndLocation>
            </DateTimeSelection>
            <OriginDestinationInformation>
            </OriginDestinationInformation>
          </DateTimeSelection>
        </OriginDestinationInformation>
        <OriginDestinationInformation>
          <OriginDestinationInformation shopIndicator="true">
            <DateTimeSelection>
              <DepartureDate>2017-05-20</DepartureDate>
              <DateTimeSelection>
                <StartLocation>LGW</StartLocation>
                <EndLocation>JFK</EndLocation>
              </DateTimeSelection>
              <OriginDestinationInformation>
                <OriginDestination elapsedTime="535" startLocation="LGW" endLocation="JFK" segmentQuantity="1">535</OriginDestination>
                <ReservationSegment segmentNumber="1" elapsedTime="535" startDateTime="2017-05-20T16:35:00" endDateTime="2017-05-20T19:30:00">535</ReservationSegment>
                <Equipment type="777">777</Equipment>
                <OperatingProviderDetails flightNumbers="2273"></OperatingProviderDetails>
              </OriginDestinationInformation>
              <OriginDestinationInformation>
                <OriginDestination elapsedTime="360" startLocation="JFK" endLocation="LGW" segmentQuantity="1">360</OriginDestination>
                <ReservationSegment segmentNumber="2" elapsedTime="360" startDateTime="2017-06-10T22:00:00" endDateTime="2017-06-11T00:00:00">360</ReservationSegment>
                <Equipment type="777">777</Equipment>
                <OperatingProviderDetails flightNumbers="2272"></OperatingProviderDetails>
              </OriginDestinationInformation>
            </OriginDestinationInformation>
          </DateTimeSelection>
        </OriginDestinationInformation>
      </DateTimeSelection>
    </OriginDestinationInformation>
  </OriginDestinationInformation>
</ExchangeShoppingRQ>
```

Below is the snippet from the response with details on a one solution and its exchange cost:

```xml
<ExchangeShoppingRS xmlns="http://services.sabre.com/sp/exchange/shopping/v2" solutions="25">
  <Solution sequence="1" pricingSequence="1">1</Solution>
  <BookItinerary>
    <OriginDestination elapsedTime="535" startLocation="LGW" endLocation="JFK" segmentQuantity="1">535</OriginDestination>
    <ReservationSegment segmentNumber="1" elapsedTime="535" startDateTime="2017-05-20T16:35:00" endDateTime="2017-05-20T19:30:00">535</ReservationSegment>
    <Equipment type="777">777</Equipment>
    <OperatingProviderDetails flightNumbers="2273"></OperatingProviderDetails>
  </BookItinerary>
  <BookItinerary>
    <OriginDestination elapsedTime="360" startLocation="JFK" endLocation="LGW" segmentQuantity="1">360</OriginDestination>
    <ReservationSegment segmentNumber="2" elapsedTime="360" startDateTime="2017-06-10T22:00:00" endDateTime="2017-06-11T00:00:00">360</ReservationSegment>
    <Equipment type="777">777</Equipment>
    <OperatingProviderDetails flightNumbers="2272"></OperatingProviderDetails>
  </BookItinerary>
  <Fare valid="true" postCalcIndex="1" pricingSequence="1" requireSplitPNR="false" passengersInDifferentCabins="false">
    <ReservationSegment segmentNumber="1"></ReservationSegment>
    <PassengerBookingDetails documentNumber="0017944161067" bookingClass="O" fareBasis="OKN8S4Z1" cabin="Y" meal="M"></PassengerBookingDetails>
  </Fare>
</ExchangeShoppingRS>
```
6.3 Partial Exchange – one leg already flown

The first leg (DFW-LAX) is already flown – ticket status shows used.

Passenger wants to change the date of the unused leg (LAX-LAX)

```
  <STL_Header.RQ ttl="300">
    <OrchestrationID seq="0">027364343467758035</OrchestrationID>
  </STL_Header.RQ>
  <POS>
    <Pseudo>G7HE</Pseudo>
    <Actual-DFW>Actual</Actual-DFW>
    <ShoppingPath requestType="TNEXC" bookingChannel="TN"/>
  </POS>
  <TicketingProvider>1S<TicketingProvider>
    <PassengerInformation>
      <PassengerWithPNR pnrLocator="DDLYBV" referenceNumber="1.1" firstName="JOHN" lastName="DOE">
```

```
<ExchangeShoppingRS xmlns="http://services.sabre.com/sp/exchange/shopping/v2" solutions="49">
  <Solution sequence="1" pricingSequence="1"/>
  <BookItinerary>
    <SeatAssignment startLocation="LAX" endLocation="DFW" directionality="FROM"/>  
  </BookItinerary>
  <ReservationSegmentDetails segmentNumber="1" meal="true" postCalcIndex="1" pricingSequence="1" requireSplitPNR="false" passengersInDifferentCabins="false"/>
  <PassengerBookingDetails documentNumber="0017944161023" bookingClass="N" fareBasis="NVAlUSN3" cabin="Y"/>
  <PassengerBookingDetails documentNumber="0017944161023" firstName="DOE" lastName="JOHN" type="ADT"/>
  <ResultPriceDifference differenceType="Refund"/>
  <FareDifference currencyCode="USD" decimalPlaces="2">875.35</FareDifference>
  <TaxDifference currencyCode="USD" decimalPlaces="2">-65.65</TaxDifference>
  <TaxDetails/>
  <SubtotalDifference currencyCode="USD" decimalPlaces="2">941.00</SubtotalDifference>
  <TotalFeeTax currencyCode="USD" decimalPlaces="2">0.00</TotalFeeTax>
  <GrandTotalDifference currencyCode="USD" decimalPlaces="2">941.00</GrandTotalDifference>
  <Passenger/>
</ExchangeShoppingRS>
6.4 Exchange for multiple passengers

```xml
<ExchangeShoppingRQ xmlns="http://services.sabre.com/sp/exchange/shopping/v2"
  xmlns:n0="http://services.sabre.com/sp/exchange/shopping/common/types/v2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.0.0">
  <STL_Header.RQ title="300">
    <OrchestrationID seq="0">02736434467758035</OrchestrationID>
  </STL_Header.RQ>
  <POS>
    <Pseudo>G7HE</Pseudo>
    <Actual>DFW</Actual>
    <ShoppingPath requestType="TNEXC" bookingChannel="TN"/>
  </POS>
  <TicketingProvider>1S</TicketingProvider>
  <PassengerInformation>
    <PassengerWithPNR pnrLocator="RQWNZB" referenceNumber="1.1" firstName="A" lastName="MARS">
      <DocumentNumber>0017944161042</DocumentNumber>
    </PassengerWithPNR>
    <PassengerWithPNR pnrLocator="RQWNZB" referenceNumber="2.1" firstName="M" lastName="KLIMAS">
      <DocumentNumber>0017944161043</DocumentNumber>
    </PassengerWithPNR>
    <PassengerWithPNR pnrLocator="RQWNZB" referenceNumber="3.1" firstName="L" lastName="PIERZCHALA">
      <DocumentNumber>0017944161044</DocumentNumber>
    </PassengerWithPNR>
    <PassengerWithPNR pnrLocator="RQWNZB" referenceNumber="4.1" firstName="J" lastName="ROMAN">
      <DocumentNumber>0017944161050</DocumentNumber>
    </PassengerWithPNR>
    <PassengerWithPNR pnrLocator="RQWNZB" referenceNumber="5.1" firstName="O" lastName="MROCKOWSKA">
      <DocumentNumber>0017944161045</DocumentNumber>
    </PassengerWithPNR>
    <PassengerWithPNR pnrLocator="RQWNZB" referenceNumber="6.1" firstName="M" lastName="DUDEK">
      <DocumentNumber>0017944161046</DocumentNumber>
    </PassengerWithPNR>
    <PassengerWithPNR pnrLocator="RQWNZB" referenceNumber="7.1" firstName="K" lastName="STAPOR">
      <DocumentNumber>0017944161047</DocumentNumber>
    </PassengerWithPNR>
    <PassengerWithPNR pnrLocator="RQWNZB" referenceNumber="8.1" firstName="G" lastName="SzcZurek">
      <DocumentNumber>0017944161048</DocumentNumber>
    </PassengerWithPNR>
    <PassengerWithPNR pnrLocator="RQWNZB" referenceNumber="9.1" firstName="W" lastName="Petliczek">
      <DocumentNumber>0017944161049</DocumentNumber>
    </PassengerWithPNR>
  </PassengerInformation>
  <OriginDestinationInformation shopIndicator="true">
    <DateTimeSelection>
      <DepartureDate>2017-06-20</DepartureDate>
    </DateTimeSelection>
    <StartLocation>LAX</StartLocation>
    <EndLocation>NRT</EndLocation>
  </OriginDestinationInformation>
  <OriginDestinationInformation shopIndicator="true">
    <DateTimeSelection>
      <DepartureDate>2017-07-01</DepartureDate>
    </DateTimeSelection>
    <StartLocation>NRT</StartLocation>
    <EndLocation>LAX</EndLocation>
  </OriginDestinationInformation>
</ExchangeShoppingRQ>
```
As you see, the exchange cost is first split per each passenger, and at the end the exchange cost is summarized.

```xml
<ExchangeShoppingRS xmlns="http://services.sabre.com/sp/exchange/shopping/v2" solutions="1">
  <Solution sequence="1" pricingSequence="1">
    <BookItinerary>
      <OriginDestination elapsedTime="635" startLocation="LAX" endLocation="NRT" segmentQuantity="1">
        <ReservationSegment segmentNumber="1" elapsedTime="635" startTime="2017-06-21T16:55:00" endTime="2017-06-21T16:55:00" startLocation="LAX" endLocation="NRT" marketingFlightNumbers="8407" marketingProvider="AA">
          <OperatingProvider operatingProvider="JL" stopQuantity="0" electronicTicketingIndicator="true" marriageGroups="O">
              <StartLocationDetails terminalID="B" GMTOffset="-9"/>
              <EndLocationDetails terminalID="B" GMTOffset="-9"/>
              <OperatingProviderDetails flightNumber="61"/>
              <DisclosureProvider disclosureProvider="JL"/>
              <ReservationSegment/>
          </OperatingProvider>
        </ReservationSegment>
      </OriginDestination>
    </BookItinerary>
    <BookItinerary>
      <OriginDestination elapsedTime="675" startLocation="NRT" endLocation="LAX" segmentQuantity="1">
        <ReservationSegment segmentNumber="2" elapsedTime="675" startTime="2017-07-01T17:25:00" endTime="2017-07-01T17:40:00" startLocation="NRT" endLocation="LAX" marketingFlightNumbers="8408" marketingProvider="AA">
          <OperatingProvider operatingProvider="JL" stopQuantity="0" electronicTicketingIndicator="true" marriageGroups="O">
              <StartLocationDetails terminalID="2" GMTOffset="-9"/>
              <EndLocationDetails terminalID="8" GMTOffset="-9"/>
              <OperatingProviderDetails flightNumber="62"/>
              <DisclosureProvider disclosureProvider="JL"/>
              <ReservationSegment/>
          </OperatingProvider>
        </ReservationSegment>
      </OriginDestination>
    </BookItinerary>
  </Solution>
</ExchangeShoppingRS>
```
<PassengerBookingDetails documentNumber="0017944161042" bookingClass="Q" fareBasis="QHY0N8Z1" cabin="Y">
  <FareComponent startLocation="NRT" endLocation="LAX" directionality="TO"/>
</PassengerBookingDetails>

<PassengerBookingDetails documentNumber="0017944161043" bookingClass="Q" fareBasis="QHY0N8Z1" cabin="Y">
  <FareComponent startLocation="NRT" endLocation="LAX" directionality="TO"/>
</PassengerBookingDetails>

<PassengerBookingDetails documentNumber="0017944161044" bookingClass="Q" fareBasis="QHY0N8Z1" cabin="Y">
  <FareComponent startLocation="NRT" endLocation="LAX" directionality="TO"/>
</PassengerBookingDetails>

<PassengerBookingDetails documentNumber="0017944161046" bookingClass="Q" fareBasis="QHY0N8Z1" cabin="Y">
  <FareComponent startLocation="NRT" endLocation="LAX" directionality="TO"/>
</PassengerBookingDetails>

<PassengerBookingDetails documentNumber="0017944161047" bookingClass="Q" fareBasis="QHY0N8Z1" cabin="Y">
  <FareComponent startLocation="NRT" endLocation="LAX" directionality="TO"/>
</PassengerBookingDetails>

<PassengerBookingDetails documentNumber="0017944161048" bookingClass="Q" fareBasis="QHY0N8Z1" cabin="Y">
  <FareComponent startLocation="NRT" endLocation="LAX" directionality="TO"/>
</PassengerBookingDetails>

<PassengerBookingDetails documentNumber="0017944161049" bookingClass="Q" fareBasis="QHY0N8Z1CH" cabin="Y">
  <FareComponent startLocation="NRT" endLocation="LAX" directionality="TO"/>
</PassengerBookingDetails>

<ReservationSegmentDetails>
  <PassengerPriceInformation>
    <Passenger documentNumber="0017944161042" type="ADT" firstName="A" lastName="MARS"/>
    <ResultPriceDifference differenceType="Refund">
      <FareDifference currencyCode="USD" decimalPlaces="2">-2055.00</FareDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2">-86.00</TaxDifference>
      <TaxDetails>
        <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="US">-0.00</TaxDifference>
        <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="YC">-0.00</TaxDifference>
        <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="XY">-0.00</TaxDifference>
        <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="XH">-0.00</TaxDifference>
        <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="AY">-0.00</TaxDifference>
        <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="SW">-0.00</TaxDifference>
        <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="OPT">-0.00</TaxDifference>
        <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="XF">-0.00</TaxDifference>
        <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="YR">-86.00</TaxDifference>
      </TaxDetails>
      <SubtotalDifference currencyCode="USD" decimalPlaces="2">-1969.00</SubtotalDifference>
      <TotalFee currencyCode="USD" decimalPlaces="2">250.00</TotalFee>
      <TotalFeeTax currencyCode="USD" decimalPlaces="2">0.00</TotalFeeTax>
      <GrandTotalDifference currencyCode="USD" decimalPlaces="2">-1719.00</GrandTotalDifference>
      <ResultPriceDifference/>
    </ResultPriceDifference>
  </PassengerPriceInformation>
</ReservationSegmentDetails>

<Passenger documentNumber="0017944161043" type="ADT" firstName="M" lastName="KLIMAS"/>
  <ResultPriceDifference differenceType="Refund">
    <FareDifference currencyCode="USD" decimalPlaces="2">-2055.00</FareDifference>
    <TaxDifference currencyCode="USD" decimalPlaces="2">-86.00</TaxDifference>
    <TaxDetails>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="US">-0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="YC">-0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="XY">-0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="XH">-0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="AY">-0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="SW">-0.00</TaxDifference>
    </TaxDetails>
  </ResultPriceDifference>
</Passenger>
<TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="Q">0.00</TaxDifference>
</TaxDetails>
</Passenger>

<Passenger documentNumber="0017944161046" type="ADT" firstName="L" lastName="PIERZCHALA">

<Refund differenceType="Refund">

<FareDifference currencyCode="USD" decimalPlaces="2">2055.00</FareDifference>
</Refund>
</Passenger>

<Passenger documentNumber="0017944161050" type="INF" firstName="J" lastName="ROMAN">

<Refund differenceType="Refund">

<FareDifference currencyCode="USD" decimalPlaces="2">206.00</FareDifference>
</Refund>
</Passenger>

<Passenger documentNumber="0017944161045" type="ADT" firstName="O" lastName="MROCKOWSKA">

<Refund differenceType="Refund">

<FareDifference currencyCode="USD" decimalPlaces="2">2055.00</FareDifference>
</Refund>
</Passenger>

<Passenger documentNumber="0017944161046" type="ADT" firstName="M" lastName="DUDEK">
<ResultPriceDifference differenceType="Refund">
  <FareDifference currencyCode="USD" decimalPlaces="2">2055.00</FareDifference>
  <TaxDetails>
    <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="US">0.00</TaxDifference>
    <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="YC">0.00</TaxDifference>
    <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="XY">0.00</TaxDifference>
    <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="A">0.00</TaxDifference>
    <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="O">0.00</TaxDifference>
    <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="XF">0.00</TaxDifference>
    <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="YR">86.00</TaxDifference>
  </TaxDetails>
  <SubtotalDifference currencyCode="USD" decimalPlaces="2">-1969.00</SubtotalDifference>
  <TotalFee currencyCode="USD" decimalPlaces="2">250.00</TotalFee>
  <GrandTotalDifference currencyCode="USD" decimalPlaces="2">-1719.00</GrandTotalDifference>
</ResultPriceDifference>

<Passenger documentNumber="0017944161047" type="ADT" firstName="K" lastName="STAPOR">
  <ResultPriceDifference differenceType="Refund">
    <FareDifference currencyCode="USD" decimalPlaces="2">2055.00</FareDifference>
    <TaxDetails>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="US">0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="YC">0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="XY">0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="A">0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="O">0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="XF">0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="YR">86.00</TaxDifference>
    </TaxDetails>
    <SubtotalDifference currencyCode="USD" decimalPlaces="2">-1969.00</SubtotalDifference>
    <TotalFee currencyCode="USD" decimalPlaces="2">250.00</TotalFee>
    <GrandTotalDifference currencyCode="USD" decimalPlaces="2">-1719.00</GrandTotalDifference>
  </ResultPriceDifference>
</Passenger>

<Passenger documentNumber="0017944161048" type="ADT" firstName="G" lastName="SZCZUREK">
  <ResultPriceDifference differenceType="Refund">
    <FareDifference currencyCode="USD" decimalPlaces="2">2055.00</FareDifference>
    <TaxDetails>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="US">0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="YC">0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="XY">0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="A">0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="O">0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="XF">0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="YR">86.00</TaxDifference>
    </TaxDetails>
    <SubtotalDifference currencyCode="USD" decimalPlaces="2">-1969.00</SubtotalDifference>
    <TotalFee currencyCode="USD" decimalPlaces="2">250.00</TotalFee>
    <GrandTotalDifference currencyCode="USD" decimalPlaces="2">-1719.00</GrandTotalDifference>
  </ResultPriceDifference>
</Passenger>

<Passenger documentNumber="0017944161049" type="CNN" firstName="W" lastName="PETLICZEK">
  <ResultPriceDifference differenceType="Refund">
    <FareDifference currencyCode="USD" decimalPlaces="2">1541.00</FareDifference>
    <TaxDetails>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="US">0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="YC">0.00</TaxDifference>
      <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="XY">0.00</TaxDifference>
    </TaxDetails>
    <SubtotalDifference currencyCode="USD" decimalPlaces="2">-1541.00</SubtotalDifference>
    <TotalFee currencyCode="USD" decimalPlaces="2">0.00</TotalFee>
    <GrandTotalDifference currencyCode="USD" decimalPlaces="2">-1541.00</GrandTotalDifference>
  </ResultPriceDifference>
</Passenger>
6.5 Not Shopped Leg

In this example the user wishes to leave the first leg (LHRPHL) untouched and only exchange the second leg (PHLLHR):

```
1.1JONES/MARK ACT
1 AA6130S 20MAY 6 LHRPHL HK1 1240 1525 /DCAA*NSSUOV /E
OPERATED BY BRITISH AIRWAYS
 2 AA6129S 26MAY 5 PHLLHR HK1 1830 0635 27MAY 6
    /DCAA*NSSUOV /E
OPERATED BY BRITISH AIRWAYS
TK1/TIME LIMIT
 1,TAW/
PHONES 1.DFW712114431
RECEIVED FROM - TD
G7HE.PSG*TD=0229/24APR17 NSSUOV II
```

Because of that, the user needs to provide full flight segment details for the non-shopped leg within ExchangeShopping request:

```
  <STL_Header.RQ title="300">
  </STL_Header.RQ>
  <OrchestrationID seq="0">027364343467758035</OrchestrationID>
  <POS>
    <Pseudo>G7HE</Pseudo>
    <Actual>DFW</Actual>
    <ShoppingPath requestType="TNEXC" bookingChannel="TN"/>
  </POS>
  <TicketingProvider>1S</TicketingProvider>
</ExchangeShoppingRQ>
```
As you can see the unchanged leg remains as it was:

<html xmlns="http://services.sabre.com/sp/exchange/shopping/v2" solutions="49">
  <bookItinerary>
    <originDestination elapsedTime="525" startLocation="LHR" endLocation="PHL" segmentQuantity="1"></originDestination>
  </bookItinerary>
</html>
<OriginDestination elapsedTime="05" startLocation="BWI" endLocation="PHL" segmentQuantity="1">
<ReservationSegment segmentNumber="3" elapsedTime="05" startDateTime="2017-04-28T07:10:00" endDateTime="2017-04-28T07:12T07:40:00" startLocation="BWI" endLocation="PHL" marketingFlightNumber="4181" marketingProvider="AA" operatingProvider="AA" stopQuantity="0" electronicTicketingIndicators="true" marriageGroup="O">
<StartLocationDetails GMTOffset="-5"/>
<EndLocationDetails GMTOffset="-5"/>
<PassengerBookingDetails documentNumber="0017944161064" bookingClass="S" fareBasis="SKW7S4C1" cabin="Y">
</PassengerBookingDetails>
</ReservationSegment>
</OriginDestination>

<OriginDestination elapsedTime="151" startLocation="PHL" endLocation="MCO" segmentQuantity="1">
<ReservationSegment segmentNumber="4" elapsedTime="151" startDateTime="2017-05-01T07:00:00" endDateTime="2017-05-01T09:31:00" startLocation="PHL" endLocation="MCO" marketingFlightNumber="2096" marketingProvider="AA" operatingProvider="AA" stopQuantity="0" electronicTicketingIndicators="true" marriageGroup="O">
<StartLocationDetails terminalID="0" GMTOffset="-5"/>
<EndLocationDetails GMTOffset="-5"/>
<PassengerBookingDetails documentNumber="0017944161064" bookingClass="S" fareBasis="SKW7S4C1" cabin="Y" meal="M">
</PassengerBookingDetails>
</ReservationSegment>
</OriginDestination>

<OriginDestination elapsedTime="603" startLocation="MCO" endLocation="LHR" segmentQuantity="1">
<ReservationSegment segmentNumber="5" elapsedTime="603" startDateTime="2017-05-11T15:37:00" endDateTime="2017-05-12T07:40:00" startLocation="MCO" endLocation="LHR" marketingFlightNumber="56" marketingProvider="AA" operatingProvider="AA" stopQuantity="0" electronicTicketingIndicators="true" marriageGroup="O">
<StartLocationDetails terminalID="3" GMTOffset="-5"/>
<StopLocation locationCode="MIA" arrivalDateTime="2017-05-11T16:45:00" departureDateTime="2017-05-11T18:00:00" elapsedTime="68" duration="75" GMTOffset="-5" equipmentType="772"/>
</ReservationSegment>
</OriginDestination>

<BookItinerary>
<Fare valid="true" postCalcIndex="1" pricingSequence="1" requireSplitPNR="false" passengersInDifferentCabin="false">
<ReservationSegmentDetails segmentNumber="1">
</ReservationSegmentDetails>
</Fare>
</BookItinerary>
<FareDifference currencyCode="USD" decimalPlaces="2">15.00</FareDifference>
<TaxDifference currencyCode="USD" decimalPlaces="2">55.70</TaxDifference>
<TaxDetails>
  <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="YR">-0.80</TaxDifference>
  <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="US">57.00</TaxDifference>
  <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="ZP">0.00</TaxDifference>
  <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="YC">0.00</TaxDifference>
  <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="XY">0.00</TaxDifference>
  <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="YA">0.00</TaxDifference>
  <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="GB">-0.30</TaxDifference>
  <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="UB">-0.20</TaxDifference>
  <TaxDifference currencyCode="USD" decimalPlaces="2" taxCode="XF">0.00</TaxDifference>
</TaxDetails>
<br />
<SubtotalDifference currencyCode="USD" decimalPlaces="2">-40.70</SubtotalDifference>
<TotalFeeTax currencyCode="USD" decimalPlaces="2">200.00</TotalFeeTax>
<br />
<GrandTotalDifference currencyCode="USD" decimalPlaces="2">240.70</GrandTotalDifference>
<br />
<Passenger>
  <PassengerPriceInformation>
    <TotalPriceDifference currencyCode="USD" decimalPlaces="2">-15.00</TotalPriceDifference>
    <TaxDifference currencyCode="USD" decimalPlaces="2">55.70</TaxDifference>
    <SubtotalDifference currencyCode="USD" decimalPlaces="2">-40.70</SubtotalDifference>
    <NonRefundableAmount currencyCode="USD" decimalPlaces="2">0.00</NonRefundableAmount>
    <TotalFee currencyCode="USD" decimalPlaces="2">200.00</TotalFee>
    <TotalFeeTax currencyCode="USD" decimalPlaces="2">0.00</TotalFeeTax>
    <GrandTotalDifference currencyCode="USD" decimalPlaces="2">240.70</GrandTotalDifference>
  </PassengerPriceInformation>
</Passenger>
<br />
</Solution>
Troubleshooting

7.1 Introduction

This section provides examples of the most common error messages, their causes, and recommended solutions where applicable.

There are few steps that Exchange Shopping makes during transaction processing and, on each of these steps, an error message can be returned if the system detects the problem.

These steps are:

- Validation of incoming request against XSD schema.
- Business validation of incoming request and documents used in the transaction.
- Further validation of the requested document(s) and itinerary is performed by underlying subsystems.
- Searching for itinerary options and validating them against Category 31 of fares in exchanged tickets, which are done by the Shopping system.
- Calculating exchange cost for passengers, which is performed by the Post Calculation process.

7.2 Error Messages by Category

The following table contains error messages (grouped in categories):

- Error messages coming from the orchestrating application

  Example:

  ```xml
  <ApplicationResults xmlns="http://services.sabre.com/STL_Payload/v02_01" status="Incomplete">
  <Error type="Application" timeStamp="2017-04-21T11:18:21.363-05:00">
    <SystemSpecificResults>
      <Message code="ERR.SP.CLIENT.INVALID_REQUEST">Incomplete POS information - missing Actual element.</Message>
    </SystemSpecificResults>
  </Error>
</ApplicationResults>
```

- Error messages coming from subsystems.

  Example:

  ```xml
  <ApplicationResults xmlns="http://services.sabre.com/STL_Payload/v02_01" status="Incomplete">
  <Error type="Application" timeStamp="2017-04-21T11:18:21.363-05:00">
    <SystemSpecificResults>
      <Message code="ERR.SP.PROVIDER_ERROR">No valid shopping combinations found.</Message>
    </SystemSpecificResults>
  </Error>
</ApplicationResults>
```
Please note that whenever the application encounters an error from any of the subsystems called the generic error message *No valid shopping combinations found* will be thrown. Service will then include all messages coming from subsystems as warnings.

### 7.2.1 Business Validation Errors

The business validation layer of Exchange Shopping request data verifies the following:

- Information in the XML request, such as checking if conflicting searching or pricing parameters were not used.
- Eligibility for the Exchange Shopping transaction checking, such as if the shopped ticket has any open for use coupons, provided reservation statuses are the ones supported by the service.

### 7.2.2.1 Table of Common Business Validation Errors

The following table provides information about some other business validation errors that you may encounter.

**Note** Long error message text contains variables (such as start and end location of leg, ticket number, and search parameters) that cause the particular issue.

<table>
<thead>
<tr>
<th>Short Error Message Text</th>
<th>Example of Long Error Message Text</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing related segments</td>
<td>Unable to process the request: MRV - SVO 2018-02-22 portion of travel was marked as not shopped but it does not contain necessary related segment(s) information. Please amend your request and try again.</td>
<td>Provide the details of air segments for not shopped leg and request again.</td>
</tr>
<tr>
<td>Missing related segments</td>
<td>Unable to process the request - for portion of travel: MRV - SVO 2018-02-23 you chose to preserve the same connecting cities but related flight segment information is missing. Required minimum number of segments specified for this shopping qualifier is 2. Please amend your request and try again.</td>
<td>Provide the details of air segments for the shopped leg to which Use Same Connections shopping parameter was applied and request again.</td>
</tr>
<tr>
<td>No origin destination has been marked for shopping</td>
<td>Invalid Exchange Shopping transaction - none of portions of travel was chosen for shopping.</td>
<td>At least one leg needs to be shopped to qualify for Exchange Shopping transaction.</td>
</tr>
<tr>
<td>Invalid combination of qualifiers</td>
<td>Invalid request - following shopping qualifiers can’t be combined together: maxStops, maxConnections. Please choose only one of them and try again.</td>
<td>Choose either Maximum Number of Stops Permitted per Leg or Maximum Number of Connections per Leg shopping parameter and request again.</td>
</tr>
<tr>
<td>Invalid combination of qualifiers</td>
<td>Invalid request - following shopping qualifiers can’t be combined together: maxStops,</td>
<td>Choose either Maximum Number of Stops Permitted per Leg or Use</td>
</tr>
<tr>
<td>Invalid combination of qualifiers</td>
<td>Invalid request - following shopping qualifiers can't be combined together: preserveConnectionIndicator, maxConnections. Please choose only one of them and try again.</td>
<td>Choose either Use Same Connections or Maximum Number of Connections Permitted per Leg shopping parameter and request again.</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Invalid combination of qualifiers</td>
<td>Invalid request - following shopping qualifiers can't be combined together: maxStops, preserveConnectionIndicator, maxConnections. Please choose only one of them and try again.</td>
<td>Choose either Maximum Number of Stops Permitted per Leg, Use Same Connections or Maximum Number of Connections Permitted per Leg shopping parameter and request again.</td>
</tr>
<tr>
<td>Invalid combination of qualifiers</td>
<td>Invalid request - following shopping qualifiers can't be combined together: onlineIndicator, interlineIndicator. Please choose only one of them and try again.</td>
<td>It is not possible to set both: interline and online service only searching parameters to &quot;true.&quot; Choose only one of them, setting it to &quot;true&quot; and request again.</td>
</tr>
<tr>
<td>Invalid combination of qualifiers</td>
<td>Invalid request - following shopping qualifiers can't be combined together: preserveConnectionsIndicator, Location. Please choose only one of them and try again.</td>
<td>Choose either Use Same Connections or Consider Only These Connection Cities/Airports shopping parameter within a single leg and request again.</td>
</tr>
<tr>
<td>Invalid combination of qualifiers</td>
<td>Invalid request - following shopping qualifiers can't be combined together: corporateID, accountCode. Please choose only one of them and try again</td>
<td>Choose either Shopping by Corporate ID or Shopping by Account Code and request again.</td>
</tr>
<tr>
<td>Same preferred and non-preferred carrier</td>
<td>Unable to process the request where the same airlines: U0, K0 are indicated as preferred and non-preferred carrier. Please amend your request and try again.</td>
<td>Decide which carrier should be your Preferred Carrier and which one should be Excluded Carrier, adjust request accordingly, and try again.</td>
</tr>
<tr>
<td>Short Error Message Text</td>
<td>Example of Long Error Message Text</td>
<td>Solution</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------</td>
<td>----------</td>
</tr>
</tbody>
</table>
| Start location mismatch   | Unable to process the request - origin airport of portion of travel: SVO - MRV does not match departure airport: LED of the first related flight segment. Please amend your request and try again. | Make sure that start location of the leg that is not shopped or is shopped but uses *Use Same Connections* shopping parameter matches departure airport of its first related segment. For more information, see the following:  
  • “Leg that Is Not Shopped,” page 21.  
  • “Leg that Is Shopped and “Use Same Connections” Shopping Parameter Is Applied,” page 23. |
| End location mismatch     | Unable to process the request - destination airport of portion of travel: SVO - MRV does not match arrival airport: PRG of the last related flight segment. Please amend your request and try again. | Make sure that end location of the leg that is not shopped or is shopped but uses *Use Same Connections* shopping parameter matches arrival airport of its last related segment. For more information, see the following:  
  • “Leg that Is Not Shopped,” page 21.  
  • “Leg that Is Shopped and “Use Same Connections” Shopping Parameter Is Applied,” page 23. |
| Flights not in correct date or time order | Check itinerary - flights not in correct date or time order. Please amend your request and try again. | Make sure that legs are sent in chronological order in Exchange Shopping request as well as the chronology of related segments. |
| Invalid - infants only request | Invalid Exchange Shopping transaction - unable to shop for: INF only. | Exchange Shopping does not support requests for infant type PTCs only. For more information, see “Requests for Infant PTCs Only,” page 35. |
| Segments discontinuity detected | Invalid request - segments discontinuity detected between: MEL and SYD in BNE - SYD portion of travel. Please amend your request forming portions of travel correctly and try again. | Please make sure there is no discontinuity (surface sector/ARNK) between related segments of given leg. If that is the case please split such a leg into two legs accordingly as surface sector is treated as stopover. |
7.3 Errors from Other Subsystems

Following are some of the common errors that may be returned by other subsystems working under Exchange Shopping. The following examples do not cover all possible errors, but list the ones that are most common.

7.3.1 Unable to Verify Interline Coupon Status

Reason
The Exchange Shopping web service was not able to get information about the real status of coupon in the “CTRL” status.

Solution
Try again in a few minutes. If the issue persists, contact the Web Services Support desk.

7.3.2 UNABLE TO REPRICE – VOLUNTARY CHANGE RULES FAILED

Reason
This error is returned when the itinerary for which the passenger shopped is not allowed by Category 31 rules of the exchanged ticket.

Solution
Verify Category 31 filing of all exchanged fares in the ticket.

7.3.3 UNABLE TO REPRICE – VOLUNTARY CHANGE RULES UNAVAILABLE

Reason
Carrier did not file Category 31 rules for some or all fares in the exchanged ticket.

Solution
Because Exchange Shopping depends on Category 31 rules processing, it is not possible to obtain exchange options if Category 31 were not filed for all fares in the exchanged ticket. You may want to review possibilities of Category 31 filing.

7.3.4 UNABLE TO MATCH FARES

The UNABLE TO MATCH FARES error is returned when it is impossible to get full information about all exchanged fares and consequently, Category 31 rules are pulled for them. There are various scenarios when this can occur. Some examples are:

- Exchanged ticket was priced manually using fares not published in the Sabre system, which is why the system cannot locate them.
- Fares in the exchanged ticket were discounted manually by more than 9 percent.
- Fares used in the exchanged ticket are not available in the Sabre system in the partition of the airline performing the Exchange Shopping transaction. (This may involve an interline ticket issued by other airline or GDS agent.)
• The exchanged ticket is a result of a previous exchange, and the ticket was partially used before the exchange (includes previously flown coupons), such as tickets exchanged by GDS agents or other airlines.
Technical Support – Travel Agencies and 3rd party developers

8.1 Introduction

If you have any questions or need assistance, contact our Sabre Dev Studio Global Customer Support Center via any of the methods provided in this chapter.

8.2 Phone

When reporting production or other critical/time sensitive issues, contact us via telephone as follows:

<table>
<thead>
<tr>
<th>Location</th>
<th>Phone Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>800-678-9460</td>
</tr>
<tr>
<td>Canada</td>
<td>682-605-5570</td>
</tr>
<tr>
<td>International</td>
<td>598-2-518-6020 or your regional Sabre Software help desk</td>
</tr>
</tbody>
</table>

8.3 Email

Email is monitored 24 x 7 with a response within 24 hours or less. The email address is: webservices.support@sabre.com

Caution Please include the Sabre Pseudo City Code (PCC) where the issue is occurring.

When reporting an issue with web services, input and output payloads are required. Please attach the payloads as separate files, and name them clearly. Samples of these files are available at Sabre Dev Studio website: https://developer.sabre.com

To help ensure that our environment is free of viruses, our policy mandates that all messages received by Sabre from external sources follow special file name guidelines. When sending zipped files please make a note of the following required file naming convention: File names must end in “.sabre.zip” or the zipped attachment will be removed by the email server (for example, “docs.zip” would need to be renamed to “docs.sabre.zip”).

If your correspondence is regarding a previously reported issue, please include the service incident (“SI”) number in the subject line of your message.