

API Developer Notes

Using Fare Quote Super Best Buy on the Apollo CRS

13 October 2011

Version 1.2

THE INFORMATION CONTAINED IN THIS DOCUMENT IS CONFIDENTIAL AND PROPRIETARY TO TRAVELPORT

Copyright

Copyright © 2011 Travelport and/or its subsidiaries. All rights reserved.

Travelport provides this document for information purposes only and does not promise that the information contained in this document is accurate, current or complete. This document is subject to change without notice.. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the licensee's personal use without the prior written permission of Travelport and/or its subsidiaries.

Trademarks

Travelport and/or its subsidiaries may have registered or unregistered patents or pending patent applications, trademarks copyright, or other intellectual property rights in respect of the subject matter of this document. The furnishing of this document does not confer any right or licence to or in respect of these patents, trademarks, copyright, or other intellectual property rights.

All other companies and product names are trademarks or registered trademarks of their respective holders.

Contents

Overview	1
Using FareQuoteSuperBB to Shop for Fares.....	2
Mapping Fares to Available Flights.....	5
Validating Availability Returned by FareQuoteSuperBB	6
Tax and Low Fare Shopping	8
Auto-Plating Carrier Logic	8
Plating Carrier Selection if Auto-Plating Carrier Logic is Not Activated.....	8
Verifying Tax Calculations if Auto-Plating Carrier Logic is Not Activated	9
Response Time Optimization.....	10
GZIP	10
Optimization.....	10
Filtering.....	12
Inhibiting of Vendors with Slow Links	12
Caching by the API Customer	12
Appendix: Related Documents	13

Overview

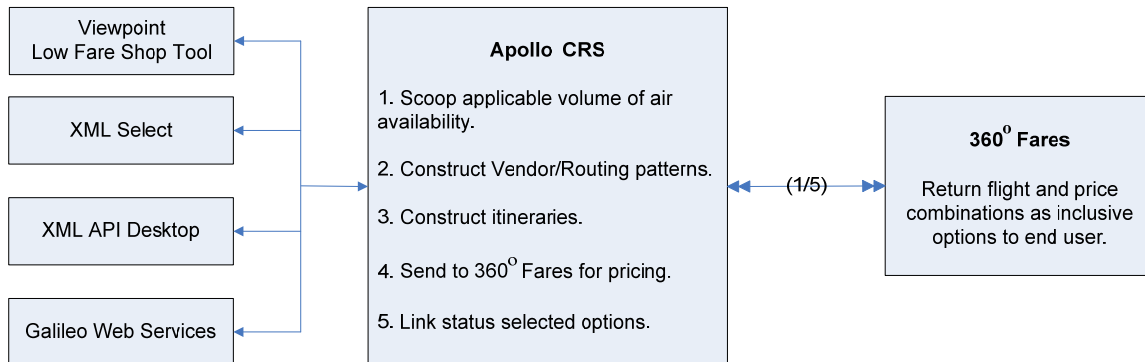
This document describes the flight and fare selection process for Low Fare Shopping, also known as Shopper Fare Quote or Super Best Buy.

Low Fare Shopping allows a user to find the best price options for a proposed journey, using origin/destination and date information, without requiring a booked itinerary. This transaction combines air availability and the fare quote request, and allows a user to enter criteria to limit the shopping request, such as time period, specific carriers to include or exclude, private fare modifiers, and passenger types.

Using FareQuoteSuperBB to Shop for Fares

The FareQuoteSuperBB_# transaction can be used by XML Select, XML API Desktop, and the XML Select service in Galileo Web Services (GWS). The CRS selects from the available flights and returns the best fares that can be obtained on these flights.

All searches send requests to the Apollo CRS and receive flight and price combinations for the end user, as shown in the following diagram.



After receiving a low fare shopping request, the Apollo CRS:

1) Scoops applicable air availability.

All flights within the range defined by the parameters of the XML request are rated to obtain the most suitable flights per origin/destination pair. The flights selected per origin/destination pair are comparable to either:

- The first flights displayed on Focalpoint if a neutral availability request is sent.
- The result of an AirAvailability_# transaction using a general availability <GenAvail> request.

The following criteria are reviewed to determine which flights are selected:

- Preference is given to non-stop and direct flights.
- Connections are rated based on overall journey time.

All flights have available inventory validated by the Apollo system using AVS or NAVS (numeric AVS) data. See the *API Developer Notes: Availability on the Apollo and Galileo CRSs* document for more information.

2) Construct Vendor/Routing Patterns

The system groups the flights per origin/destination by a unique vendor and routing combination. For example, a journey from London to New York can be traveled in a number of ways, including the following examples.

Origin/Destination 1: LON to NYC

BA: LHR-JFK (pattern 1)
BA0117

AF: LHR-CDG-JFK (pattern 5)
AF1871 / AF0003

BA0175	AF1877 / AF0005
BA0177	
BA: LHR-EWR (pattern 2)	BA: LHR-BOS / AA: BOS-JFK (pattern 6)
BA0151	BA0213 / AA4662
BA0152	
AA: LHR-JFK (pattern 3)	
AA0105	
AA0101	
AA0107	
CO: LGW-EWR(pattern 4)	
CO0019	
CO0029	

Origin/Destination 2: NYC to LON

BA: JFK-LHR (pattern 1)	AF: JFK-CDG-LHR (pattern 4)
BA0118	AF0004 / AF1872
BA0176	AF0006 / AF1878
BA0178	
AA: JFK-LHR (pattern 2)	AA: JFK-BOS / BA: BOS-LHR (pattern 5)
AA0106	AA4663 / BA0214
AA0102	
AA0108	
CO: EWR-LGW (pattern 3)	
CO0020	
CO0030	

The patterns are ranked from most appropriate to least appropriate, based on internally defined parameters, to determine which patterns may provide the most convenient and inexpensive fares.

3) Construct Itinerary Options

At this stage, the patterns returned for the individual origin/destinations are combined to create all possible itinerary options (for one-way searches the patterns **are** the itinerary options). For example, an itinerary option can be:

Outbound Origin/Destination 1 LON to NYC

BA: LHR-JFK (pattern 1)
 BA0117
 BA0175
 BA0177

Return Origin/Destination 2 NYC to LON

BA: JFK-LHR (pattern 1)
 BA0118
 BA0176
 BA0178

Each individual itinerary option is evaluated and ranked as appropriate, based on overall journey time and overall class availability. The process is repeated until the final itinerary options are returned.

4) 360° Fares Pricing

The itineraries that are selected are sent to the faring process. Only the lowest price is returned for any single given itinerary option. SuperBB does not provide multiple fares such as refundable and non-refundable for the same set of flights. Looking again at the example in *Construct Itinerary Options*:

BA: LHR-JFK

BA0117

BA0175

BA0177

BA: JFK-LHR

BA0118

BA0176

BA0178

The lowest fare (Fare 1) for this itinerary is calculated. If all of the associated flights are not available at the *lowest* fare, the Apollo CRS discards the remaining flights that are not available at the lowest fare.

For example:

If BA has 10 flights between DUB and LHR on a given day, and only six flights are available in the lowest booking code, the Apollo CRS will not quote the remaining four flights.

5) Link Status Selected Options

A percentage of flights per origin/destination pair are validated for last seat availability using vendor links. Solutions returned by FareQuoteSuperBB have an indicator to identify the status of the solutions that have been retrieved across the vendor link.

The Inside Availability link provides Apollo with direct link to the vendor's database. The indicator in the <InsideAvailOption> element can have the following values to indicate the Inside Availability status of the response:

Indicator Value	Description
Y	The Inside Availability option exists.
N	No Inside Availability option exists.
L	The Inside Availability option was used on this flight. This flight <i>has</i> been polled.
P	Inside Availability was used on this flight, and availability was obtained on a point-to-point basis.
M	Inside availability was used on this flight, and availability was obtained on an origin and destination basis. The carrier <i>has not</i> implemented Journey Control.
J	Inside Availability was used on this flight, availability and was obtained on an origin and destination basis. The carrier <i>has</i> implemented Journey Control.

Mapping Fares to Available Flights

For every origin/destination pair within the FareQuoteSuperBB_# request, there is a corresponding <AirAvail> element in the FareQuoteSuperBB_# response. For every fare available, there is a <FareInfo> element in the response.

Within the <FareInfo> element, the <FlightItemCrossRef> element correlates each fare to its applicable flights. See the *API Developer Notes: Low Fare Shopping*.

Validating Availability Returned by FareQuoteSuperBB

If necessary, you can confirm the availability response returned by the Low Fare Shopping transaction either:

- Before the results display to the end user.
- After the user completes flight selection, and before the information gathering and booking process.

Occasionally, errors occur as a result of inaccurate AVS/NAVS data. However, there are two reasons that you may NOT want to confirm the response:

1) Excessive Transaction Charges (ETAs):

Each customer has a Galileo look-to-book ratio outlined in their contract. For every request to the CRS beyond the defined look-to-book ratio, an ETA is applied. It is in the customer's best interest to minimize the number of transactions to the host.

2) Airline look-to-book ratio:

Each linked availability request sent to a vendor system has a cost implication to the vendor. If a Pseudo City Code (PCC) sends excessive link status messages to vendors and does not complete enough bookings, that PCC has a high look-to-book ratio. If a PCC has an excessively high look-to-book ratio, vendors may "switch off" the ability to link status availability for that PCC. If a vendor switches off your link status availability, you will receive poorer responses from the Low Fare Shopping request as all availability will be polled via AVS/NAVS.

If availability must be confirmed before it is displayed to the end user, apply the following rules and logic to minimize the impact to the vendor system and to the CRS:

- 1) The Low Fare Shopping transaction validates a subset of flights per origin/destination pair. Validated flights are indicated by presence of 'L','P','M' or 'J' in the <InsideAvailOption> element and **do not** require confirmation before displaying to the end user.
- 2) Confirming Last Seat Availability for flight options returned by Low Fare Shopping responses can be limited. For example, the five lowest fares can be returned, or those fare options that are **less than 100%** more expensive than the lowest fare can be returned.

For example, if a low fare of €100 is returned and a high fare of €2000 is returned, it is not typically necessary to confirm the €2000 fare as it is unlikely that this fare will be selected by the end user.

- 3) Any flight options that are eligible for validation after the previous two items are taken into consideration should be validated/queried for Last Seat Availability using Apollo in the most efficient manner:

- **Inside Availability/Seamless Availability Participants**

Inside Availability/Seamless Availability participants opt to receive interactive (linked) availability messages from Apollo, even if the user specifically asks for General or Neutral availability. Therefore, validation of availability for seamless availability participants can be sufficiently achieved using AirAvailability_# with <GenAvail> and <InhibitLinkStatus> set to N.

- **NAVS Participants**

As a general rule, carriers that supply Apollo with numeric AVS (NAVS) data update this data based on demand. As a result, data stored in the Apollo system should be sufficiently accurate to operate with an acceptable booking failure rate and does not require validation.

- **Standard AVS Participant**

A direct request needs to be sent using AirAvailability_# with the <AirVSpecificAvail> element for Standard AVS participants.

For more information, see the *API Developer Notes: Availability on the Apollo and Galileo CRSs* document.

Tax and Low Fare Shopping

In the FareQuoteSuperBB_# response, a particular fare can be applicable for several outbound and several inbound flight options. As a result, tax calculations are not completely accurate because some taxes depend on plating carrier. *Plating carrier* is a travel industry term that refers to having the authority from an airline to issue tickets for that airline. It is a legacy term from the time when airlines were issued actual plates to print hard-copy tickets.

For example, for many carriers, the YQ/YR (fuel surcharge) tax, as well as other taxes, do not display unless a plating carrier is specified. Although the plating carrier can be specified in the FareQuoteSuperBB_# request, it is only be practical for a search involving one airline.

Because the taxes associated with FareQuoteSuperBB_# responses cannot always be accurately calculated, the Apollo CRS was enhanced with the addition of auto-plating carrier logic.

Auto-Plating Carrier Logic

Auto-plating carrier logic evaluates the itineraries in the FareQuoteSuperBB_# response and, using various rules, selects a plating carrier for each itinerary. In this process:

1. The AAT AUTH field of the PCC specified for ticketing is checked.
 - If the selected plating carrier exists in the AUTH table and has an interline agreement with all other carriers in the itinerary, then that plating carrier is used.
 - If the carrier is not included in the AUTH field of the Pseudo City Code (PCC) specified for ticketing, another carrier is selected to be the plating carrier, based on additional rules.
2. The Apollo CRS evaluates all carriers in the itinerary, if necessary, and only then does it display the error: *No Availability for this request*. This error may be misleading, but it usually indicates that auto-plating carrier logic is activated and the relevant carriers are not defined in the AUTH field of the PCC.

The auto-plating carrier logic is activated per PCC. Contact your account manager or local NDC/SMO to discuss activating the auto-plating carrier logic if it is not already activated. If you do not perform the ticketing yourself, you need to add the PCC of the ticketing agent into the TKAG field of the AAT of your own PCC. Contact your local NDC/SMO for this configuration.

Note: Fare quotes are only guaranteed when stored and/or ticketed.

Plating Carrier Selection if Auto-Plating Carrier Logic is Not Activated

The AAT table of your PCC defines the plating carriers for which you are authorized. If the auto-plating carrier logic is not activated, rules are followed to select a plating carrier. These rules should be verified by your local Billing Settlement Plan (BSP), as plating rules vary by BSP.

The rules for plating carrier selection are similar to the following rules. However, these rules are not universal, and must be verified with each BSP. The customer system used to select the plating carrier should also be submitted to that BSP for approval if it writes tickets.

Note: Any BSP Airline may, by written notice to Agents, restrict the use of that airline's CIP (Carrier Identification Plate). This rule means that you must allow for the possibility of exceptions to whatever rule is agreed upon with the BSP.

The general rules on which the BSP bases their requirement are:

- a) For transportation relating to the following types of journey,
- Wholly within a country, or
 - Wholly between Canada and the USA, or
 - Wholly within the area comprising Denmark, Norway, Sweden, or
 - Wholly within the area consisting of Bahrain, Oman, Qatar and the United Arab Emirates, or
 - Wholly within the area consisting of Brunei, Malaysia and Singapore.

The ticketing airline shall be the airline (or its General Sales Agent at the place of ticket issuance) performing the first sector of the transportation, or, if this is not possible, any Billing Settlement Plan Airline participating in the transportation, or the General Sales Agent at the place of ticket issuance of any airline participating in the transportation.

- b) For transportation relating to other types of journey, the ticketing airline shall be any Billing Settlement Plan Airline participating in the transportation, or a Billing Settlement Plan Airline acting as General Sales Agent for the country of ticket issuance of any airline participating in an international sector of the transportation.

Verifying Tax Calculations if Auto-Plating Carrier Logic is Not Activated

The following transactions can be used to verify tax calculations, although the use of these transactions impacts the customers' ETAs (Excessive Transaction Charges).

- FareQuoteClassSpecific_# returns the fare and tax information for the exact booking code specified, regardless of availability.
- FareQuoteFlightSpecific_# with <InhibitLinkStatus> set to 'Y' returns the cheapest fare within the specified cabin class, including accurate tax information, and checks availability against AVS/NAVS data.
- FareQuoteFlightSpecific_# with <InhibitLinkStatus> set to 'N' returns the cheapest fare within the specified cabin class, including accurate tax information, and checks availability against AVS/NAVS data unless the vendor is a 100% seamless or a partially seamless participant, in which case availability is polled across the vendor link.

Response Time Optimization

To ensure optimization of response times for a FareQuoteSuperBB_# request, the following options can be applied or considered:

- GZIP
- Optimization
- Filtering
- Inhibiting of vendors with slow links
- Caching by the API customer

GZIP

Galileo Web Service customers must apply GZIP data compression.

GZIP (GNU zip) is a compression utility that uses algorithms to compress data. GZIP reduces the size of the named files using Lempel-Ziv coding (LZ77). Whenever possible, each file is replaced by another file with the extension .gz, while keeping the same ownership modes, access, and modification times.

GWS uses GZIP to compress requests to the GWS Gateway and responses from the GWS Gateway. The request for a GZIP response is defined in the HTTPS header.

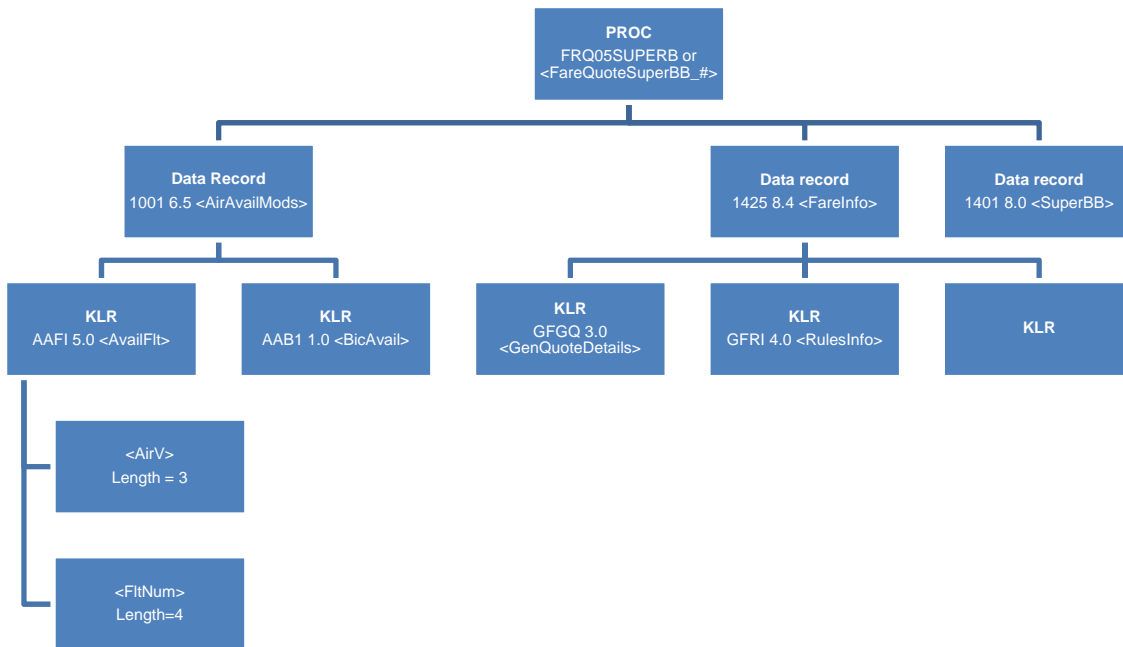
Optimization

Optimization of the FareQuoteSuperBB_# request ensures that unnecessary data is not processed or returned by the host. The benefits of optimization are:

- Faster processing time on the host.
- A reduced data payload returned across the Internet.

Structured data procedures (PROCS) are the structured data equivalent of an XML transaction. PROCS are constructed of one or more Data Records, which are constructed of one or more Key Logical Records (KLR), which are constructed of many data fields.

The following diagram illustrates how PROCS, Data Records, KLRs, and data fields are related. Note that the diagram contains a subset of KLRs available in the FareQuoteSuperBB_# transaction; while the diagram only shows two KLRs within the 1425 Data Record, there are actually over 70 KLRs in the 1425 Data Record.



To implement optimization, the XML request identifies the necessary KLRs to be returned and instructs the host to only return these KLRs in the response. The <Optimize> element in the XML request indicates exactly which Data Records should be returned and, within those Data Records, exactly which KLRs should be returned. All data fields within the specified KLRs are returned.

In the following example, the <Optimize> element instructs the CRS to process and return only the nine KLRs listed, rather than processing and returning the over 70 KLRs that are available in this transaction. By using the <Optimize> element, unnecessary CRS processing is alleviated, and the data sent across the Internet is reduced.

```

<SuperBBMods>
  <Optimize>
    <RecType>1001</RecType> <---Record Type 1001 is <AirAvail>-->
    <KlrIDary>
      <KlrID>AAFI</KlrID> <---<AvailFlt>--->
      <KlrID>AAB1</KlrID> <---<BicAvail>--->
    </KlrIDary>
  </Optimize>
  <Optimize>
    <RecType>1425</RecType> <---Record Type 1425 is <FareInfo>-->
    <KlrIDary>
      <KlrID>GFGQ</KlrID> <---GFGQ is <GenQuoteDetails>--->
      <KlrID>GFXI</KlrID> <---GFXI is <FlightItemCrossRef>--->
      <KlrID>GFPI</KlrID> <---GFPI is <PassengerType>--->
      <KlrID>GFRP</KlrID> <---GFRP is <RulesData>--->
      <KlrID>GFRI</KlrID> <---GFRI is <RulesInfo>--->
      <KlrID>GFPX</KlrID> <---GFPX is <EnhancedPrivateFares>-->
      <KlrID>GRFB</KlrID> <---GRFB is <FareBasisCodeSummary>-->
    </KlrIDary>
  </Optimize>

```

Filtering

Filtering optimizes data at the Structured Data (SD) to XML translator and, therefore, does not reduce CRS processing time. However, filtering provides the following benefits:

- Reduces the data payload sent back to the client across the Internet/network for Galileo Web Services. For XML Select and XML API Desktop, the performance gain is minimal because the XML translator is on the client side.
- Simplifies the XML that the client has to handle.

Filters indicate which parts of the response you want the SD translator to return. Unlike optimization, which only allows you to define Data Records and KLRs to be returned, filtering allows you to define the response the down to data field level.

For example, in a Hotel Availability response, if you want to view only the property list, use the following filter:

```
<HotelAvailability_#>
  <HotelAvailabilityPropList>
    <_ xmlns=""/>
  </HotelAvailabilityPropList>
</HotelAvailability_#>
```

However, if you want to only display the chain code, RoomMasterID, property name, address, and BIC codes, use the following filter:

```
<HotelAvailability_#>
  <HotelAvailabilityPropList>
    <ChainCodeAry>
      <ChainCode>
        <Chain></Chain>
        <RoomMasterID></RoomMasterID>
        <PropName></PropName>
        <ShortAddr></ShortAddr>
        <BICAry>
          <_ xmlns=""/>
        </BICAry>
      </ChainCode>
    </ChainCodeAry>
  </HotelAvailabilityPropList>
</HotelAvailability_#>
```

Inhibiting of Vendors with Slow Links

The entire FareQuoteSuperBB_# response is delayed until a vendor link response is received for all 18 polled flights. When a specific air carrier generates insignificant sales volumes, and that carrier delays the return of FareQuoteSuperBB_# responses due to slow link responses, it is possible to Inhibit Link Staging (polling) for Low Fare Shopping for that carrier. This situation is difficult to determine and should only be examined as a last resort.

Caching by the API Customer

Caching may be applied by the API Customer as determined by the customer.

Appendix: Related Documents

See the following API Developer Notes for information that can be used in conjunction with the *Using Fare Quote Super Best Buy on the Apollo CRS* document.

Document	Description
<i>Availability on the Apollo and Galileo CRS</i>	Explains how availability works on the Apollo and Galileo CRSs.
<i>Creating a PNR</i>	Explains how to create a PNR or Booking File for Air.
<i>Low Fare Air Shopping</i>	Describes in general how the FareQuoteSuperBB_# transactions can be used for low fare shopping.
<i>Using Fare Quote Super Best Buy on the Galileo CRS</i>	Explains how FareQuoteSuperBB_# functions on the Galileo CRS.