API Developer Notes

Using Fare Quote Super Best Buy on the Galileo CRS

29 June 2012

Version 1.3

Contents

Overview

This document describes the flight and fare selection process for Low Fare Shopping, also known as Shopper Fare Quote or Super Best Buy, where *More Solutions* and *Intelligent Polling* are activated.

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.

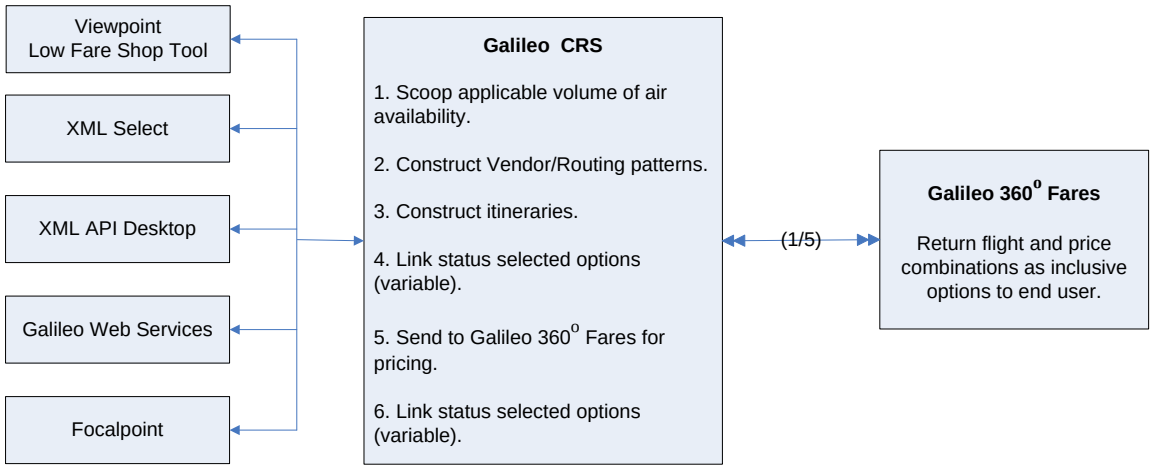
More Solutions refers to the recent enhancement where the number of fares that are returned has increased. *Intelligent Polling* refers to the improved logic that is used to determine the flights that are selected for polling across the vendor link. Both More Solutions and Intelligent Polling are only available on the Galileo CRS.

The document is specific to the most common setup for the majority of markets; however, there may be slight variations of the heuristic being used depending on your location. Contact your Galileo Account manager to see if More Solutions and Intelligent Polling are activated for your PCC.

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 Galileo 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 Galileo 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:

   - Non-stop and direct flight are preferable.
   - Connections are rated based on overall journey time.

   All flights have available inventory validated by the Galileo system using AVS or NAVS (numeric AVS) data. See the *API Developer Notes: Availability on the Apollo and Galileo CRSs* 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)	AF: LHR-CDG-JFK (pattern 5)
BA0117	AF1871 / AF0003

BA0175 BA0177	AF1877 / AF0005
BA: LHR-EWR (pattern 2) BA0151 BA0152	BA: LHR-BOS / AA: BOS-JFK (pattern 6) BA0213 / AA4662
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. **Link Status Selected Options Before Pricing**

   See *Intelligent Polling* on *page 6*.

5. Galileo 360o Fares Pricing

The itineraries that are selected are sent through the faring process. Only the lowest price is returned for any single given itinerary option. FareQuoteSuperBB_# does not provide multiple fares, such as refundable and non-refundable fares, for the same set of flights. Looking again at the example in *Step 3. 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. However, if all of the associated flights are not available at the *lowest* fare, the Galileo CRS attempts to price and return the second and third lowest fares to present all the above flight options. The CRS provides a pricing solution for all flights in the itinerary. It is the activation of More Solutions that forces the calculation of the second and third lowest fare if necessary.

For example:

If British Airways has 10 flights between DUB and LHR on a given day, and only six flights are available in the lowest booking code, More Solutions attempts to recover the four flights that are unavailable at the single lowest fare in the BA DUB-LHR itinerary group.

The Galileo CRS quotes the remaining four flights using the second lowest fare. Note that this process does not filter the six flights that *were* available at the lowest fare. Instead, the flight option is displayed again, if available, at the second lowest fare. The process is repeated again if the four flights are not available at the second lowest fare. A scenario similar to the following example can be generated:

Fare 1 (€100.00): BA112, BA114, BA116, BA118, BA120, BA122

Fare 2 (€120.00): BA112, BA114, BA116, BA118, BA120, BA122, BA124, BA126, BA128

Fare 3 (€140.00): BA112, BA114, BA116, BA118, BA120, BA122, BA124, BA126, BA128, BA130

If all 10 flights are available at the lowest fare, the above scenario does not occur because all flights in the group/itinerary have been quoted at the lowest possible fare. In the context of a “low fare search,” all objectives have been met and no further processing is required.

Note also that More Solutions *does not* return additional pricing options if the second lowest fare is more than 100% more expensive than the lowest option.

6. Link Status Selected Options After Pricing

See *Intelligent Polling* on *page 6*.

Intelligent Polling

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 indicator in <InsideAvailOption> can have the following values:

Indicator Value	Description
Y	Inside Availability option exists.
N	No Inside Availability option exists.
L	Inside Availability option was used on this flight. This flight *has* been polled.
P	Inside Availability was used on this flight and was obtained on a point-to-point basis.
M	Inside Availability was used on this flight and was obtained on an origin and destination basis. Carrier *has not* implemented Journey Control.
J	Inside Availability was used on this flight and was obtained on an origin and destination basis. Carrier *has* implemented Journey Control.

Existing Intelligent Polling Options

Polling Before Pricing (IBP)

Applicable flights per origin/destination pair are polled across the vendor link (Step 4 in *Using FareQuoteSuperBB to Shop for Fares* on *page 2*). Because this polling path is invoked prior to pricing, fare information is not available. Therefore, it is not possible to specifically target the lowest fare. However, the best flights from a class availability point of view are polled and their definite availability is considered in the faring process.

Polling After Pricing (IAP)

Applicable flights per origin/destination pair are polled across the vendor link (Step 6 in *Using FareQuoteSuperBB to Shop for Fares* on *page 2*) after the pricing process. If a carrier link response indicates that the booking code used to construct the pricing response is *unavailable*, the flight is discarded and not returned to the user. However, the flight may display if not all associated flights were available at the *lowest* fare.

Polling Before and After Pricing (IPS)

The IPS option is a combination of IBP and IAP. The total flights polled per origin/destination pair include half of the flights polled at Step 4 in *Using FareQuoteSuperBB to Shop for Fares* on *page 2* (before pricing) and half of the flights polled at Step 6 in *Using FareQuoteSuperBB to Shop for Fares* on *page 2* (after pricing).

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

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 host 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 'L', 'P', 'M', or 'J' in <InsideAvailOption>, 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 have been taken into consideration should be validated/queried for Last Seat Availability using Galileo in the most efficient manner:

   - **Inside Availability/Seamless Availability Participants**

     Inside Availability/Seamless Availability participants opt to receive interactive (linked) availability messages from Galileo, 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**

     Generally, carriers that supply Galileo with numeric AVS (NAVS) data update this data based on demand. As a result, data stored in the Galileo 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 <AirVSpecificAvail> 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 Galileo CRS was enhanced with the addition of the auto-plating carrier logic.

Auto-Plating Carrier Logic

The 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 Galileo 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.

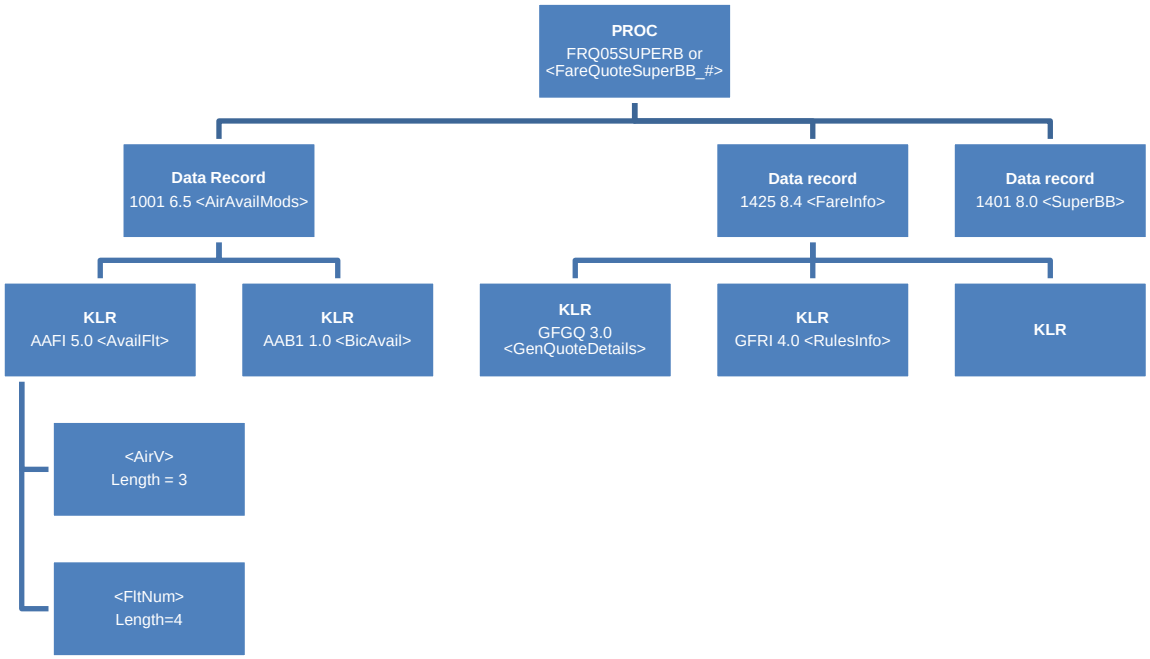
Optimization

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, <Optimize> 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 <Optimize>, 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.
- 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 polled flights. When a specific air carrier generates insignificant sales volumes and that carrier delays the return of the FareQuoteSuperBB_# response due to slow link responses, it is possible to Inhibit Link Statusing (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 Galileo CRS* document:

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