



Air Availability Data

User Guide

Table of Contents

1. Introduction	3
1.1. Product Objective	3
1.2. Benefits Statement:	3
2. Glossary.....	3
3. Product Scope	4
4. Functional Requirements Overview.....	4
4.1. Configurable Parameters	4
4.2. Scope of Data	4
4.3. Message Structure	5
4.4. Data Transfer Requirements.....	8
5. Set Up Process.....	8

1. Introduction

1.1. Product Objective

The objective of this product is to provide air schedules and availability data for selected markets and travel dates to customers in a batch mode in XML message format via secure FTP.

Air Availability Data has access to air schedules and availability data by consuming a Sabre Web Service called AirSchedulesAndAvailability (ASAA).

Requests to this service are built based on parameters that can be configured for each customer.

The frequency of the data capture and the data file transmission can also be specified for each customer.

1.2. Benefits Statement:

Air Availability Data (AAD) is a revenue driver that differentiates Sabre from competition. It allows customers to leverage air schedules and availability data to populate their own caches, thereby enhancing their product. The data can be used to refresh online travel agencies and developers own airline inventory content. AAD data is delivered via bundled data sets.

2. Glossary

XML	Extensible Markup Language is a text-based format used to share data on the World Wide Web, intranets, and elsewhere.
FTP	File Transfer Protocol is the standard network protocol used for the transfer of computer files between a client and server on a computer network.
ASAA	AirSchedulesAndAvailability is a Sabre Web Service.
PCC	Pseudo City Code is an alpha-numeric identifier for a corporate user of a computer reservation system (CRS) or global distribution system (GDS), typically a travel agency.
O&D	Origin and Destination pair
Solution	A solution may contain a single direct flight or a connection between the origin and the destination.
EFG	Sabre Enhanced FTP Gateway

Gzip	It is a file format and a software application used for file compression and decompression.
Booking Code	Cabin class of service, also known as Reservation Booking Designator (RBD) or IATA Class Code.

3. Product Scope

Sabre Air Schedules and Availability data returned by AirSchedulesAndAvailability Web Service is filtered out and forwarded in a batch mode according to the parameters and frequency defined for the customer.

4. Functional Requirements Overview

4.1. Configurable Parameters

In order to build and submit requests to ASAA service, AAD needs the following parameters to be defined for each customer:

- PCC – at least one is mandatory
- List of markets (O&Ds) – mandatory
- Travel date range (from date; to date or advance purchase days) – optional; default is today's date for 7 days
- Included carriers – optional
- Excluded carriers – optional
- Number of solutions per market (max. 300) – optional; default is 100
- Maximum number of segments per solution (1 to 9) – optional; default is 3

4.2. Scope of Data

Air Availability Data data consists of solutions for a given market (O&D) and travel date. A solution may have information of a direct flight segment or connecting segments as follow:

- Origin airport code
- Destination airport code
- Departure date and time
- Arrival date and time
- Marketing carrier and flight number

- Operating carrier and flight number
- List of booking classes with corresponding numeric availability (seats) and status*

*Availability statuses:

- CR = Booking class is closed; possible to wait list
- CL = Booking class is closed; wait list is open
- CC = Booking class is closed; wait list is closed

4.3. Message Structure

AAD delivers Sabre Air Schedules and Availability data in XML message format (data files are compressed GZIP).

XML Message Schema:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="MarketAvailabilityView">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Request">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="Origin"/>
              <xs:element type="xs:string" name="Destination"/>
              <xs:element type="xs:string" name="TravelDate"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Response">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Solution" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Segment" maxOccurs="unbounded" minOccurs="0">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element type="xs:string" name="Origin"/>
                          <xs:element type="xs:string" name="Destination"/>
                          <xs:element type="xs:string" name="DepartureDateTime"/>
                          <xs:element type="xs:string" name="ArrivalDateTime"/>
                          <xs:element type="xs:string" name="MarketingCarrier"/>
                          <xs:element type="xs:string" name="MarketingFlightNumber"/>
                          <xs:element type="xs:string" name="OperatingCarrier"/>
                          <xs:element type="xs:string" name="OperatingFlightNumber"/>
                          <xs:element name="Availability">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="BookingClass" maxOccurs="unbounded" minOccurs="0">
                                  <xs:complexType>
                                    <xs:simpleContent>
                                      <xs:extension base="xs:string">
                                        <xs:attribute type="xs:string" name="code" use="required"/>

```


Example of a XML message:

```

<MarketAvailabilityView>
  <Request>
    <Origin>JFK</Origin>
    <Destination>KRR</Destination>
    <TravelDate>2017-06-25</TravelDate>
  </Request>
  <Response>
    <Solution>
      <Segment>
        <Origin>JFK</Origin>
        <Destination>FRA</Destination>
        <DepartureDateTime>2017-06-25T11:00:00</DepartureDateTime>
        <ArrivalDateTime>2017-06-25T20:00:00</ArrivalDateTime>
        <MarketingCarrier>LH</MarketingCarrier>
        <MarketingFlightNumber>4841</MarketingFlightNumber>
        <OperatingCarrier>LH</OperatingCarrier>
        <OperatingFlightNumber>4841</OperatingFlightNumber>
        <Availability>
          <BookingClass code="A" seats="9" />
          <BookingClass code="B" seats="9" />
          <BookingClass code="C" seats="9" />
          <BookingClass code="D" seats="9" />
          <BookingClass code="E" seats="9" />
          <BookingClass code="F" seats="9" />
          <BookingClass code="G" seats="9" />
          <BookingClass code="H" seats="9" />
          <BookingClass code="J" seats="9" />
          <BookingClass code="K" seats="9" />
          <BookingClass code="L" seats="9" />
          <BookingClass code="M" seats="9" />
          <BookingClass code="N" seats="9" />
          <BookingClass code="P" seats="9" />
          <BookingClass code="Q" seats="9" />
          <BookingClass code="S" seats="9" />
          <BookingClass code="T" seats="9" />
          <BookingClass code="U" seats="9" />
          <BookingClass code="V" seats="9" />
          <BookingClass code="W" seats="9" />
          <BookingClass code="Y" seats="9" />
          <BookingClass code="Z" seats="9" />
        </Availability>
      </Segment>
      <Segment>
        <Origin>FRA</Origin>
        <Destination>KRR</Destination>
        <DepartureDateTime>2017-06-25T21:50:00</DepartureDateTime>
        <ArrivalDateTime>2017-06-25T23:20:00</ArrivalDateTime>
        <MarketingCarrier>LH</MarketingCarrier>
        <MarketingFlightNumber>1370</MarketingFlightNumber>
        <OperatingCarrier>LH</OperatingCarrier>
        <OperatingFlightNumber>1370</OperatingFlightNumber>
        <Availability>
          <BookingClass code="B" seats="9" />
          <BookingClass code="C" seats="9" />
          <BookingClass code="D" seats="9" />
          <BookingClass code="H" seats="9" />
          <BookingClass code="J" seats="9" />
          <BookingClass code="K" seats="0" status="CC" />
          <BookingClass code="L" seats="0" status="WL" />
          <BookingClass code="M" seats="9" />
        </Availability>
      </Segment>
    </Solution>
  </Response>
</MarketAvailabilityView>

```

```
<BookingClass code="P" seats="7"/>
<BookingClass code="Q" seats="9"/>
<BookingClass code="S" seats="9"/>
<BookingClass code="T" seats="9"/>
<BookingClass code="U" seats="9"/>
<BookingClass code="V" seats="9"/>
<BookingClass code="W" seats="9"/>
<BookingClass code="Y" seats="9"/>
<BookingClass code="Z" seats="9"/>
</Availability>
</Segment>
</Solution>
</Response>
</MarketAvailabilityView>
```

4.4. Data Transfer Requirements

Output files containing the air schedules and availability data are pushed to customers using secure file transfer protocol (SFTP). Gzip is used to compress the data files. The data may be split into multiple files if too much data for a single file.

5. Set Up Process

1. Customer fills out the connectivity details in the EFG data collector attached to this document
2. Sabre supplies FTP account information for CERT (non-prod environment)
3. Validation testing is performed in CERT (non-prod environment)
4. Sabre supplies FTP account information for PROD
5. Validation testing is performed in PROD



EFG_Data_Collector_
outgoing.docx