



2001C Message Specification

Abstract

This document presents the specifications for the Air Availability RQ/RS message pairs, the Package Tours RQ/RS message pairs, and the Golf Tee Times RQ/RS message pairs.

Section 1 - The Air Working Group presents the Air Availability section with descriptions and definitions of XML data components.

Section 2 – This section presents the Package Tours RQ/RS message pairs.

Section 3 – This section presents the Golf Tee Times RQ/RS message pairs.

OpenTravel Alliance, Inc.
333 John Carlyle Street, Suite 600
Alexandria, Va. 22314 USA
+1 703-548-7005
Fax +1 703-548-1264
opentravel@disa.org
<http://www.opentravel.org/>

Prepared in partnership with Data Interchange Standards Association (<http://www.disa.org>)

OpenTravel™ Alliance, Inc.
License Agreement
Copyright, OpenTravel Alliance (OTA), 2001.
All rights reserved, subject to the User License set out below.

Authorization to Use Specifications and documentation

IMPORTANT: The OpenTravel™ Alliance ("OTA") Message Specifications ("Specifications"), whether in paper or electronic format, are made available subject to the terms stated below. Please read the following carefully as it constitutes a binding Agreement, based on mutual consideration, on you and your company as licensee ("You").

- 1. Documentation.** OTA provides the Specifications for voluntary use by individuals, partnerships, companies, corporations, organizations, and other entities at their own risk. The Specifications and any OTA supplied supporting information, data, or software in whatever medium in connection with the Specifications are referred to collectively as the "Documentation."
- 2. License Granted.**
 - 2.1.** OTA holds all rights, including copyright, in and to the Documentation. OTA grants to You this perpetual, non-exclusive license to use the Documentation, subject to the conditions stated below. All use by You of the Documentation is subject to this Agreement.
 - 2.2.** You may copy, download, and distribute the Documentation and may modify the Documentation solely to allow for implementation in particular contexts. You may bundle the Documentation with individual or proprietary software and/or sublicense it in such configurations.
 - 2.3.** You must reference, in a commercially reasonable location, the fact that the OTA Documentation is used in connection with any of your products or services, in part or in whole, whether modified or not, and You may include truthful and accurate statements about Your relationship with OTA or other use of the Documentation.
 - 2.4.** However, you may not change or modify the Specification itself, develop a new standard or specification from the Documentation, or state or imply that any works based on the Documentation are endorsed or approved by OTA.
 - 2.5.** You must include the OTA copyright notice in connection with any use of the Documentation. Any uses of the OTA name and trademarks are subject to the terms of this Agreement and to prior review and approval by OTA.
 - 2.6.** Nothing in this Agreement shall be interpreted as conferring on You or any other party any other interest in or right to the Documentation. Nothing in this Agreement shall be interpreted as in any way reducing or limiting OTA's rights in the Documentation.
- 3. LIABILITY LIMITATIONS.** *THIS AGREEMENT IS SUBJECT TO THE FOLLOWING LIABILITY LIMITATIONS:*
 - 3.1.** ANY DOCUMENTATION PROVIDED PURSUANT TO THIS NON-EXCLUSIVE LICENSE AGREEMENT IS PROVIDED "AS IS" AND NEITHER OTA NOR ANY PERSON WHO HAS CONTRIBUTED TO THE CREATION, REVISION, OR MAINTENANCE OF THE DOCUMENTATION MAKES ANY REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.
 - 3.1.** Neither OTA nor any person who has contributed to the creation, revision or maintenance of the documentation makes any representations or warranties, express or implied, that the use of the documentation or software will not infringe

any third party copyright, patent, patent application, trademark, trademark application, or other right.

- 3.2.** Neither OTA nor any person who has contributed to the creation, revision, or maintenance of the documentation shall be liable for any direct, indirect, special or consequential damages or other liability arising from any use of the documentation or software. You agree not to file a lawsuit, make a claim, or take any other formal or informal action against OTA based upon Your acquisition, use, duplication, distribution, or exploitation of the Documentation.
 - 3.3.** The foregoing liability limitations shall apply to and be for the benefit of OTA, any person who has contributed to the creation, revision or maintenance of the documentation, and any member of the board of directors, officer, employee, independent contractor, agent, partner, or joint venturer of OTA or such person.
- 4. No Update Obligation.** Nothing in this Agreement shall be interpreted as requiring OTA to provide You with updates, revisions or information about any development or action affecting the Documentation.
- 5. No Third Party Beneficiary Rights.** This Agreement shall not create any third party beneficiary rights.
- 6. Application to Successors and Assignees.** This Agreement shall apply to the use of the Documentation by any successor or assignee of the Licensee.
- 7. Term.** The term of this license is perpetual subject to Your compliance with the terms of this Agreement, but OTA may terminate this License Agreement immediately upon your breach of this Agreement and, upon such termination you will cease all use duplication, distribution, and/or exploitation of the Documentation in any manner.
- 8. Interpretation and Choice of Forum.** The law of the Commonwealth of Virginia and any applicable Federal law shall govern this Agreement. Any disputes arising from or relating to this Agreement shall be resolved in the courts of the Commonwealth of Virginia, including Federal courts. You consent to the jurisdiction of such courts and agree not to assert before such courts any objection to proceeding in such forum.
- 9. Acceptance.** Your acceptance of this License Agreement will be indicated by your affirmative acquisition, use, duplication, distribution, or other exploitation of the Documentation. If you do not agree to these terms, please cease all use of the Documentation now.
- 10. Questions.** Questions about the Agreement should be directed to www.opentravel.org.

Table of Contents

Section 1. Flight Availability and Point Of Sale	5
1.1. Existing EDI-based systems	5
1.2. XML in place of EDI	5
1.3. Flight Availability	6
1.4. Root verbs	6
1.5. <OTA_AirAvailRQ>	6
1.6. <OTA_AirAvailRS>	9
1.7. Sample Messages	11
Section 2. – Package Tours/Holiday Bookings	15
2.0. Package Availability Request	15
2.1. Package Availability Use Case	16
2.2. Package Availability Messages	16
2.3. Create Booking	28
2.4. Create Booking Use Case	28
2.5. Create Booking Messages	28
Section 3. – Golf Tee Times	35
3.1. Golf Course Search Request Message (OTA_CourseSearchRQ)	36
3.2. Golf Course Search Response Message (OTA_CourseSearchRS)	36
3.3. Course Availability Request Message (OTA_CourseAvailRQ)	37
3.4. Course Availability Response Message (OTA_CourseAvailRS)	39
3.5. Course Tee Time Reservation Request Message (OTA_CourseResRQ)	41
3.6. Course Tee Time Reservation Response Message (OTA_CourseResRS)	43
3.7. Sample XML Message for Course Search Request	43
3.8. Sample XML Message for Course Search Response	44
3.8.1. Sample XML Message for Course Availability Request	45
3.9. Sample XML Message for Course Availability Response	45
3.10. Sample XML Message for Course Reservation Request	45
3.11. Sample XML Message for Course Reservation Response	47
3.12. Revision History	48

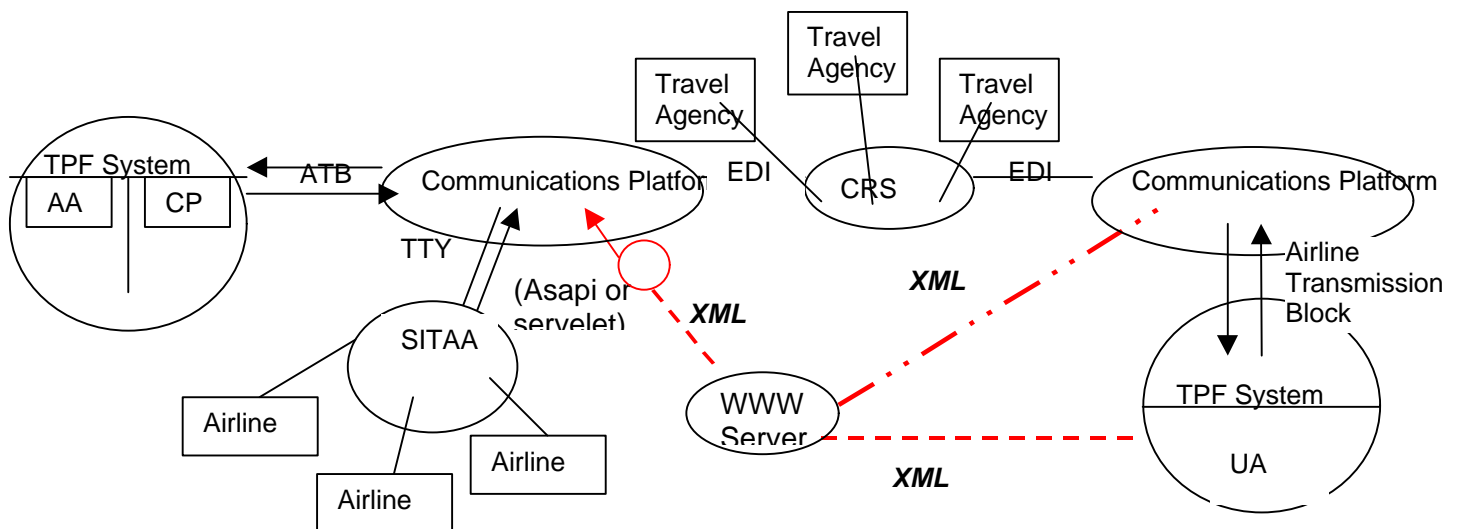
Section 1. Flight Availability and Point Of Sale

This specification addresses the structure, elements, and context of requests and responses for airline flight availability and point of sale information. It is intended to provide a simple interface to Air Reservation Systems for querying availability in 3 scenarios:

1. General availability request for multiple available flight options from multiple carriers.
2. Specific availability request for multiple flight options from a single carrier.
3. Specific availability request for a specific flight from a specific carrier.

1.1. Existing EDI-based systems

It is presumed that airlines currently use EDI as the “common language” to exchange data. The example below shows how XML can be used in place of EDI when two suppliers communicate:



A partitioned TPF System serving multiple airlines (AA and CP here) sends an airline transmission block, which is proprietary to each carrier system. These transmission blocks carry messages in uncommon formats. The communications platform acts as a translator, and sends messages to other entities into a commonly agreed upon language with specific structures and business rules. This language, currently in use by the airline industry, is called EDI.

Translated messages move from the communications platform to external entities such as CRSs and SITA, who in turn distribute them to multiple airlines. The diagram above shows UA as the eventual supplier recipient; the UA communications platform translates the EDI message into the language used by its TPF system.

EDI is currently dependent upon dedicated lines and third-party distributors, both of which add expense to the data transmission process. XML enables messages to be transmitted from suppliers to travel agents, other suppliers and third parties over the Internet, enabling more flexibility in communication and lowering communications costs.

1.2. XML in place of EDI

The ideal concept would be to send ATBs directly from a TPF system to the communications platform using XML, which would not require translation at the communications platform level. If a carrier is not currently supporting EDI, then creating the ability to send and receive XML

messages directly into its TPF system could fairly straightforward; some airlines are currently developing this capability.

A faster solution requiring less investment for EDI-translating systems would be to add an additional block to the communications platform to convert airline transmission blocks into XML. If a carrier prefers to outsource its XML services, the communications layer can continue to export its data blocks in EDI, but can designate the XML outsourcer as a recipient. The outsourcer could then translate and distribute the XML messages as necessary. In all three cases, the XML would be sent to a www server, which in turn sends the XML message to an asapi or servlet that sends the XML-formatted messages to the participating carrier for translation back into the TPF system.

Systems may be slowed down with each additional translation layer, so minimization of stops along the way is ideal, and can possibly be achieved by reaching agreements with trading partners to establish direct Internet links.

1.3. Flight Availability

The airline flight availability messages listed in this section are designed to allow a system to query a supplier system using XML as the common language rather than EDI. The querying system identifies itself, requests that the supplier system search for airline flights and flight services, and then the supplier may return which flights meet the defined criteria. Presumably, an availability request is made with the intent to ultimately book a flight reservation.

1.4. Root verbs

The single OTA_AirAvailRQ schema definition supports three types of flight availability requests: specific flight, specific airline, and multiple airline. A specific flight request seeks availability information on particular flights on particular dates. A specific airline request is more general, seeking availability for a journey that might occur during a range of times or dates. A multiple airline request is one that combines one or more dates or segments that are not normally priced for sale as a unit.

The following two schemas are defined:

OTA_AirAvailRQ – The request message. Seeks availability of flights on particular date.

OTA_AirAvailRS – The response message. Determines availability within the constraints of specified criteria for flights on particular dates.

1.5. <OTA_AirAvailRQ>

This request seeks availability of flights on particular dates for an origin–destination pair from one or more suppliers. The request consists of the following child elements:

Element	MinOccurs	MaxOccurs
"POS"	1	1
"OriginDestinationInformation"	1	1
"SpecificRequestInformation"	0	1
"TravelPreferences"	0	1
"TravelInformation"	0	1

The OTA_AirAvailRQ element contains 2 attributes:

Attribute	Type	Use	Choices	Default
"DesiredOptionQuantity"	xs:positiveInteger	optional		10
"DirectFlightsOnly"	xs:positiveInteger	optional		False

1.5.1. <POS>

This is the standard OTA point of sale information element. It is detailed in the 2001C Specification schema fragments.

1.5.2. <OriginDestinationInformation>

This element contains the information relating to the travel origin and destination, plus information on times and dates of travel and connection points. It may contain the following child elements:

Element	MinOccurs	MaxOccurs
"OriginLocation"	1	1
"DestinationLocation"	1	1
"TravelDateTime"	1	1
"Connection"	0	1

The OriginDestinationInformation element has no attributes.

1.5.3. <OriginLocation>, <DestinationLocation>

OriginLocation and DestinationLocation are elements of type LocationType, a complex Type based on xs:string which has 2 attributes:

Attribute	Type	Use	Choices	Default
"LocationCode"	xs:string	Required		
"CodeContext"	xs:string	Optional	"IATA", "Internal"	"IATA"

1.5.4. <TravelDateTime>

TravelDateTime is an element of type TravelDateTime, a complexType with 4 attributes:

Attribute	Type	Use	Choices	Default
"TravelDate"	xs:dateTime	Required		
"TravelTimeIndicator"	xs:string	Optional	"DepartureTime", "ArrivalTime"	"DepartureTime"
"TimeWindowBefore"	xs:duration	Optional		
"TimeWindowAfter"	xs:duration	Optional		

1.5.5. <Connection>

Connection is an element of type ConnectionType. It has one child element, ConnectionLocation which is of type LocationType:

Element	MinOccurs	MaxOccurs
"ConnectionLocation"	1	1

Connection has a single attribute which indicates the preference level of the requested connection:

Attribute	Type	Use	Choices	Default
"PreferLevel"	PreferLevelType	Optional	"Only",	"Preferred"

“Acceptable”,
“Preferred”

1.5.6. <SpecificRequestInformation>

The SpecificRequestInformation element is used to restrict the request to a specific airline, specific flight or specific booking class. It has 3 child elements and no attributes:

Element	MinOccurs	MaxOccurs
“FlightNumber”	0	1
“Airline”	0	1
“BookingClassPref”	0	1

FlightNumber is of type xs:positiveInteger.

1.5.7. <Airline>

Airline is an element of type CompanyName, a complex type based on xs:string with 2 attributes:

Attribute	Type	Use	Choices	Default
“CompanyCode”	xs:string	required		
“CodeContext”	xs:string	optional	“IATA”, “internal”	“IATA”

1.5.8. <BookingClassPref>

BookingClassPref is an element with 2 attributes:

Attribute	Type	Use	Choices	Default
“BookingClassCode”	ResBookDesignCodeType	required	Single alphabet character	
“CodeContext”	PreferLevelType	optional	“Only”, “Unacceptable”, “Preferred”	“Preferred”

1.5.9. <TravelPreferences>

The TravelPreferences element holds preference information related to the flight search. The child elements are based on the Air Preferences section of the OTA 2001A Profiles Specification:

Element	MinOccurs	MaxOccurs
“VendorPref”	0	Unbounded
“FlightTypePref”	0	Unbounded
“AirFareRestrictPref”	0	Unbounded
“AirEquipPref”	0	Unbounded
“AirCabinPref”	0	Unbounded
“AirSvcClassPref”	0	Unbounded
“TicketDistribPref”	0	Unbounded

These elements are documented fully in the 2001A Specification.

1.5.10. <TravelerInformation>

The TravelerInformation element allows the type and number of proposed travelers to be included in the Availability Request message. It contains a single element and no attributes:

Element	MinOccurs	MaxOccurs
"PassengerTypeQuantity"	1	Unbounded

1.5.11. <PassengerTypeQuantity>

PassengerTypeQuantity is an element with 2 attributes used to specify a Passenger Type Code (PTC), plus the number of passengers who match that PTC:

Attribute	Type	Use	Choices	Default
"PassengerTypeCode"	ThreeLetterCodeType	Required		
"Quantity"	xs:positiveInteger	Required		

1.6. <OTA_AirAvailRS>

This response message details availability of flights on particular dates for an origin–destination pair from one or more suppliers. The response consists of the following child elements:

Element	MinOccurs	MaxOccurs
"OTA_SucceedDef"	0	1
"OTA_FailureDef"	0	1
"OriginDestinationOption"	0	Unbounded

The OTA_AirAvailRS element has no attributes. Either OTA_SucceedDef or OTA_FailureDef must be present. They are defined in

1.6.1. <OriginDestinationOption>

An OriginDestinationOption element is the container for a set of connected flight segments that originate at the OriginLocation specified in the OTA_AirAvailRQ message and terminate at the DestinationLocation.

It has no attributes and 1 child element:

Element	MinOccurs	MaxOccurs
"FlightSegment"	1	Unbounded

1.6.2. <FlightSegment>

The FlightSegment Element represents a single flight with a single flight number. It can include stopovers.

It has the following child elements:

Element	MinOccurs	MaxOccurs
"DepartureAirport"	1	1
"ArrivalAirport"	1	1
"DepartureDateTime"	1	1
"ArrivalDateTime"	1	1
"Cabin"	0	Unbounded
"MarketingAirline"	1	1
"OperatingAirline"	0	1
"Equipment"	0	1
"Comment"	0	Unbounded

It has the following attributes:

Attribute	Type	Use	Choices	Default
"SequenceNumber"	xs:integer	Required		
"FlightNumber"	xs:positiveInteger	Required		
"StopQuantity"	xs:nonNegativeInteger	Optional		
"JourneyDuration"	xs:duration	Optional		
"SmokingInd"	SmokingIndType	Optional	"Yes", "No"	"No"
"OnTimeRate"	OnTimeRateType	Optional	<i>Percentage(1 to 90)</i>	
"Ticket"	TicketType	Optional	"Paper", "eTicket"	"Paper"

SequenceNumber is a reference place holder to identify this particular flight segment and must be unique within this message.

1.6.3. <DepartureAirport>,<ArrivalAirport>

DepartureAirport and ArrivalAirport are elements of type LocationType, a complex Type based on xs:string which has 2 attributes:

1.6.4. <DepartureDateTime>, <ArrivalDateTime>

DepartureDateTime and ArrivalDateTime are elements of type xs:dateTime. They have no attributes.

1.6.5. <Cabin>

The Cabin element is a construct for holding a set of Booking classes, plus the seat availability for each class within a particular aircraft cabin (First, Business or Economy). It also contains attributes to identify the cabin and give meal information for the cabin.

Element	MinOccurs	MaxOccurs
"BookingClassAvail"	0	Unbounded

Attribute	Type	Use	Choices	Default
"CabinType"	CabinType	Required	"First", "Business", "Economy"	
"MealCode"	MealCodeType	Optional	Single Alphabet Character	

1.6.6. <MarketingAirline>, <OperatingAirline>

MarketingAirline and OperatingAirline are both elements of type CompanyName, a complex type based on xs:string with 2 attributes:

Attribute	Type	Use	Choices	Default
"CompanyCode"	xs:string	Required		
"CodeContext"	xs:string	Optional	"IATA", "Internal"	"IATA"

Company Code in both cases should be a 2 letter airline code.

1.6.7. <Equipment>

Equipment is an element of type EquipmentType, a complex type based on xs:string with 1 attribute:

Attribute	Type	Use	Choices	Default
"AirEquipType"	xs:string	Required	3 letter aircraft type	

1.6.8. <Comment>

Comment is an element of type xs:string for storing any text information relating to this flight segment.

1.6.9. <BookingClassAvail>

BookingClassAvail is an element of type BookingClassAvailabilityType, a complex type based on xs:string with 2 attributes:

Attribute	Type	Use	Choices	Default
"ResBookDesigCode"	ResBookDesigCode	Required	Single Alphabet Character	
"Availability"	xs:string	Required		

While Availability will generally be a numeric value indicating the number of seats available for a specific booking class, it is defined as a string to allow non-numeric indicators to be included such as waitlisted, available, etc. It is envisaged that separate attributes will be defined for these in the future.

1.7. Sample Messages

1.7.1. OTA_AirAvailRQ

1.7.1.1. General Availability Request

This is a general availability request for flights between London and Los Angeles for 2 adults and 1 child.

```
<OTA_AirAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="OTA_AirAvailRQ.xsd" DesiredOptionQuantity="10" DirectFlightsOnly="false">
  <POS>
    <Source AgentSine="BSIA1234PM" PseudoCityCode="2U8" ISOCountry="US" ISOCurrency="USD">
      <UniqueURL URL="http://provider1.org/OTAEngine/" Type="OTA_AirAvailRQ" Id="ID0507G4325"
Instance="2001-06-03T13:09:21"/>
    </Source>
  </POS>
  <OriginDestinationInformation>
    <OriginLocation LocationCode="LON" CodeContext="IATA"/>
    <DestinationLocation LocationCode="LAX" CodeContext="IATA"/>
    <TravelDateTime TravelDate="2001-09-11T09:00:00" TravelTimeIndicator="DepartureTime"
TimeWindowBefore="P3D" TimeWindowAfter="P3D"/>
  </OriginDestinationInformation>
  <TravelPreferences SmokingInd="No" MaxStopsQuantity="1">
    <VendorPref PreferLevel="Preferred">
      <VendorName CompanyCode="BA" CodeContext="IATA"/>
    </VendorPref>
    <FlightTypePref PreferLevel="Preferred">Direct</FlightTypePref>
    <AirEquipPref PreferLevel="Preferred" AirEquipType="757"/>
    <AirCabinPref PreferLevel="Preferred" CabinType="Economy"/>
    <TicketDistribPref PreferLevel="Only" DistribType="Mail" TicketTime="P5D"/>
  </TravelPreferences>
  <TravelerInformation>
    <PassengerTypeQuantity PassengerTypeCode="ADT" Quantity="2"/>
  </TravelerInformation>
</OTA_AirAvailRQ>
```

```

    <PassengerTypeQuantity PassengerTypeCode="CNN" Quantity="1"/>
  </TravelerInformation>
</OTA_AirAvailRQ>

```

1.7.1.2. Specific Airline Availability Request

This is a specific airline availability request for flights between Miami and Los Angeles on American Airlines for 1 adult with a preferred connection in Dallas.

```

<OTA_AirAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="OTA_AirAvailRQ.xsd" DesiredOptionQuantity="10" DirectFlightsOnly="false">
  <POS>
    <Source AgentSine="BSIA1234PM" PseudoCityCode="2U8" ISOCountry="US" ISOCurrency="USD"
CarrierCode="AA" AirportCode="IAD" FirstDepartPoint="IAD" ERSPUserId="12345432">
      <UniqueId URL="http://provider1.org/OTAEngine/" Type="OTA_AirAvailRQ" Id="ID0507G4325"
Instance="2001-06-03T13:09:21"/>
      <BookingChannel Type="GDS" Primary="1"/>
    </Source>
  </POS>
  <OriginDestinationInformation>
    <OriginLocation LocationCode="MIA" CodeContext="IATA"/>
    <DestinationLocation LocationCode="LAX" CodeContext="IATA"/>
    <TravelDateTime TravelDate="2001-09-11T09:00:00" TravelTimeIndicator="DepartureTime"/>
    <Connection PreferLevel="Preferred">
      <ConnectionLocation LocationCode="DFW" CodeContext="IATA"/>
    </Connection>
  </OriginDestinationInformation>
  <SpecificRequestInformation>
    <Airline CompanyCode="AA" CodeContext="IATA"/>
  </SpecificRequestInformation>
  <TravelerInformation>
    <PassengerTypeQuantity PassengerTypeCode="ADT" Quantity="1"/>
  </TravelerInformation>
</OTA_AirAvailRQ>

```

1.7.1.3. Specific Flight Availability Request

This is a specific flight request for availability in booking class Q on American Airlines flight 123 between JFK and Los Angeles for 1 adult.

```

<OTA_AirAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="OTA_AirAvailRQ.xsd">
  <POS>
    <Source AgentSine="BSIA1234PM" PseudoCityCode="2U8" ISOCountry="US" ISOCurrency="USD"
CarrierCode="AA">
      <UniqueId URL="http://provider1.org/OTAEngine/" Type="OTA_AirAvailRQ" Id="ID0507G4325"
Instance="2001-06-03T13:09:21"/>
    </Source>
  </POS>
  <OriginDestinationInformation>
    <OriginLocation LocationCode="JFK" CodeContext="IATA"/>
    <DestinationLocation LocationCode="LAX" CodeContext="IATA"/>
    <TravelDateTime TravelDate="2001-09-11T09:00:00"/>
  </OriginDestinationInformation>
  <SpecificRequestInformation>
    <Airline CompanyCode="AA" CodeContext="IATA"/>
    <FlightNumber>123</FlightNumber>
    <BookingClassPref BookingClassCode="Q" PreferLevel="Only"/>
  </SpecificRequestInformation>
  <TravelerInformation>
    <PassengerTypeQuantity PassengerTypeCode="ADT" Quantity="1"/>
  </TravelerInformation>
</OTA_AirAvailRQ>

```

1.7.2. OTA_AirAvailRS

1.7.2.1. Successful Response

This is a successful response to a general availability request. The specific requests have exactly the same format, but would not return options outside the restrictions specified in the request.

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_AirAvailRS xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="OTA_AirAvailRS.xsd">
  <OriginDestinationOption>
    <FlightSegment SequenceNumber="1" FlightNumber="212" StopQuantity="0"
JourneyDuration="P0Y0M0DT12H30M" SmokingInd="No" OnTimeRate="90" Ticket="Paper">
      <DepartureAirport LocationCode="LHR" CodeContext="IATA"/>
      <ArrivalAirport LocationCode="LAX" CodeContext="IATA"/>
      <DepartureDateTime>2001-09-11T10:30:00+00:00</DepartureDateTime>
      <ArrivalDateTime>2001-09-11T11:14:00+08:00</ArrivalDateTime>
      <Cabin CabinType="First" MealCode="L">
        <BookingClassAvail ResBookDesigCode="F" Availability="9"/>
      </Cabin>
      <Cabin CabinType="Business" MealCode="L">
        <BookingClassAvail ResBookDesigCode="C" Availability="7"/>
        <BookingClassAvail ResBookDesigCode="J" Availability="7"/>
      </Cabin>
      <Cabin CabinType="Economy" MealCode="S">
        <BookingClassAvail ResBookDesigCode="Q" Availability="0"/>
        <BookingClassAvail ResBookDesigCode="V" Availability="4"/>
        <BookingClassAvail ResBookDesigCode="N" Availability="1"/>
        <BookingClassAvail ResBookDesigCode="Y" Availability="7"/>
      </Cabin>
      <MarketingAirline CompanyCode="BA" CodeContext="IATA"/>
      <Equipment AirEquipType="744"/>
    </FlightSegment>
  </OriginDestinationOption>
  <OriginDestinationOption>
    <FlightSegment SequenceNumber="2" FlightNumber="112" StopQuantity="0"
JourneyDuration="P0Y0M0DT12H00M" SmokingInd="No" OnTimeRate="90" Ticket="Paper">
      <DepartureAirport LocationCode="LHR" CodeContext="IATA"/>
      <ArrivalAirport LocationCode="LAX" CodeContext="IATA"/>
      <DepartureDateTime>2001-09-11T10:20:00+00:00</DepartureDateTime>
      <ArrivalDateTime>2001-09-11T11:13:50+08:00</ArrivalDateTime>
      <Cabin CabinType="First" MealCode="L">
        <BookingClassAvail ResBookDesigCode="F" Availability="6"/>
      </Cabin>
      <Cabin CabinType="Business" MealCode="L">
        <BookingClassAvail ResBookDesigCode="C" Availability="4"/>
        <BookingClassAvail ResBookDesigCode="J" Availability="4"/>
      </Cabin>
      <Cabin CabinType="Economy" MealCode="S">
        <BookingClassAvail ResBookDesigCode="Q" Availability="4"/>
        <BookingClassAvail ResBookDesigCode="V" Availability="4"/>
        <BookingClassAvail ResBookDesigCode="N" Availability="2"/>
        <BookingClassAvail ResBookDesigCode="Y" Availability="6"/>
      </Cabin>
      <MarketingAirline CompanyCode="UA" CodeContext="IATA"/>
      <Equipment AirEquipType="777"/>
    </FlightSegment>
  </OriginDestinationOption>
  <OriginDestinationOption>
    <FlightSegment SequenceNumber="3" FlightNumber="22" StopQuantity="0"
JourneyDuration="P0Y0M0DT6H30M" SmokingInd="No" OnTimeRate="90" Ticket="Paper">
      <DepartureAirport LocationCode="LHR" CodeContext="IATA"/>
      <ArrivalAirport LocationCode="JFK" CodeContext="IATA"/>
      <DepartureDateTime>2001-09-11T09:30:00+00:00</DepartureDateTime>
```

```
<ArrivalDateTime>2001-09-11T11:35:00+05:00</ArrivalDateTime>
<Cabin CabinType="First" MealCode="L">
  <BookingClassAvail ResBookDesigCode="F" Availability="9"/>
</Cabin>
<Cabin CabinType="Business" MealCode="L">
  <BookingClassAvail ResBookDesigCode="C" Availability="7"/>
  <BookingClassAvail ResBookDesigCode="J" Availability="7"/>
</Cabin>
<Cabin CabinType="Economy" MealCode="S">
  <BookingClassAvail ResBookDesigCode="Q" Availability="0"/>
  <BookingClassAvail ResBookDesigCode="V" Availability="4"/>
  <BookingClassAvail ResBookDesigCode="N" Availability="1"/>
  <BookingClassAvail ResBookDesigCode="Y" Availability="7"/>
</Cabin>
<MarketingAirline CompanyCode="BA" CodeContext="IATA"/>
<Equipment AirEquipType="767"/>
<Comment>Departs Terminal 4</Comment>
</FlightSegment>
<FlightSegment SequenceNumber="4" FlightNumber="4632" StopQuantity="0"
JourneyDuration="P0Y0M0DT4H30M" SmokingInd="No" OnTimeRate="90" Ticket="Paper">
  <DepartureAirport LocationCode="JFK" CodeContext="IATA"/>
  <ArrivalAirport LocationCode="LAX" CodeContext="IATA"/>
  <DepartureDateTime>2001-09-11T14:00:00+05:00</DepartureDateTime>
  <ArrivalDateTime>2001-09-11T16:35:00+08:00</ArrivalDateTime>
  <Cabin CabinType="Business" MealCode="L">
    <BookingClassAvail ResBookDesigCode="C" Availability="7"/>
    <BookingClassAvail ResBookDesigCode="J" Availability="7"/>
  </Cabin>
  <Cabin CabinType="Economy" MealCode="S">
    <BookingClassAvail ResBookDesigCode="Q" Availability="0"/>
    <BookingClassAvail ResBookDesigCode="V" Availability="0"/>
    <BookingClassAvail ResBookDesigCode="N" Availability="0"/>
    <BookingClassAvail ResBookDesigCode="Y" Availability="7"/>
  </Cabin>
  <MarketingAirline CompanyCode="BA" CodeContext="IATA"/>
  <OperatingAirline CompanyCode="US" CodeContext="IATA"/>
  <Equipment AirEquipType="757"/>
</FlightSegment>
</OriginDestinationOption>

<OTA_SucceedDef>
  <Success/>
</OTA_SucceedDef>
</OTA_AirAvailRS>
```

Section 2. – Package Tours/Holiday Bookings

A **package holiday** usually consists of a single “pre-defined” offering with or without a choice of basic elements such as transport and accommodation. The business model for this concept is that allocated blocks of transport and accommodation inventory for a ‘season’ or ‘brochure period, typically ‘Summer’ (May to October) and ‘Winter’ (November to April) are reserved by a tour operator from the supplier. These are combined into package holiday inventory items, and set up and sold from the tour operator’s system. Notification to the original supplier of the take-up of individual inventory items takes place a short period before departure of the customers. The use cases covered in this document relate to the selling by the tour operator of the packages from their internal inventory stock.

A booking can contain any number of itinerary elements, such as transport, accommodation, car rental, extra products or services, special services, extras etc. Itinerary or journey elements are distinct by type of service and product, place of delivery, date and time the service is offered and can be individually assigned to one or more of the customers involved in the booking.

The parties involved in the current business interactions comprise Travel Agents (on behalf of customers) making enquiries and bookings with the Tour Operators who publish brochures describing the package tours on offer. The normal interaction medium is videotex which, due to the limited screen display size (80 characters x 25 lines), requires a considerable number of message pairs to achieve a booking. However, it is well-established and extensively used, with some operators taking the majority of their bookings this way.

This document covers two scenarios – Package Availability and Package Booking. The Availability phase checks a selected package against the supplier’s system and provides full details and costings and the Booking phase completes the cycle by committing the customer to paying for the holiday and the supplier to providing it. Each scenario can be invoked independently of the others, subject to the necessary minimum information being supplied in the request message.

Package Availability comprises the following messages :.	
Package Availability Request.	OTA_PkgAvailRQ
Package Availability Response	OTA_PkgAvailRS
Package Booking comprises the following messages :	
Package Booking Request.	OTA_PkgBookRQ
Package Booking Response	OTA_PkgBookRS

2.0. Package Availability Request

The Package Availability Request message is designed to establish whether a specific package is available for a specific date and duration for a given number of customers (who may be subdivided by category e.g. Adult, Child etc.).

If the request is satisfied, the enquirer will be provided with a priced breakdown of the package elements.

If the request is not satisfied because one or more elements of the package are not available, the enquirer may be provided with a selection of alternatives for that element.

2.1. Package Availability Use Case

The business use case that supports this message identifies a customer or agent (person or system acting on behalf of the customer) who requests the availability status of a specific occurrence of a package. The first step in this use case is for the enquirer to supply the details of the package, the stay and the party composition.

The steps of the use case proceed as follows:

- The Customer or agent requests the availability of a specific package for a date and duration for a number of passengers.
- The system returns a priced package summary detailing all possible combinations of facilities (where appropriate).

The data returned at Step 2 is used as the basis for the Package Booking Request.

Additional data that accompanies the response message may include information which may affect the enquirer's decision on whether to book the package, e.g. building works, unavailable facilities etc.

Where the supplier system is unable to provide costs for all combinations, it may return a basic priced summary with details of the availability of facilities from which the customer must make a choice and submit a revised request in order to get a full costing.

Possible business processing errors include:

- One or more components of the package cannot satisfy the number of passengers for the date and duration requested. The system may return a list of possible alternative components and if the enquirer chooses one from the list as a substitute the use case will restart from step 1.

Scenario

A customer wants to know if the package consisting of the Hotel Miramar in Alcudia Majorca travelling on a specific return flight pair between London Gatwick and Palma is available for 2 adults and 2 children for 14 nights from 02 October 2001. The supplier system responds with information of the flight, the hotel and prices for single, twin, triple bedded rooms for variable valid occupancies which could be generated by 2 adults and 2 children (e.g. one twin with two extra beds or a single for 1 adult with a triple for 1 adult and two children etc).

2.2. Package Availability Messages

The specification for Package Availability provides a request/response pair of messages to support the functionality of enquiring for the availability of a package and its facilities, together with responses to cover the functionality of providing suitable alternatives should the requested package not be available.

The following action verbs are used as root elements of a payload document to obtain package availability data:

- **OTA PkgAvailRQ** – Provides the identification criteria for the package and customer requirements.
- **OTA PkgAvailRS** - Returns a priced summary of the requested package or Warnings, or Errors if the request did not succeed.

2.2.1. Package Availability Request

The <OTA_PkgAvailRQ> message identifies the required package and availability criteria to be met.

The root element of the <OTA_PkgAvailRQ> contains the standard payload attributes found in all OTA payload documents as well as the attribute <ReqRespVersion> that requests a specific version of the response message. As this is the first publication of the OTA package availability message set, currently the only valid value is "1". The other components are <POS> which identifies the requester, and <PackageRequest> which identifies the package required and the party makeup.

Package Request

<PackageRequest> identifies the package to be satisfied and defines the date, duration and customers by age category.

The components of <PackageRequest> are:

- *PackageType* – describes a type of TTL product, e.g. flight only, air package etc.
- *TourCode* – the supplier's code to identify the tour or accommodation arrangement for the package.
- *Travel Code* - the supplier's code to identify the travel arrangement for the package.
- *CompanyCode* – identifies the supplier of the package.
- *BrochureCode* – identifies the brochure in which the package is offered (Optional).
- *BoardCode* – identifies the preferred board terms (Optional).
- *SKU* (Stock Keeping Unit) – an alternative method of identifying the full tour package
- *URL* (Uniform Resource Locator) – another alternative method of identifying the full tour package

DateRange – defines the start date and duration of the required package.

InventoryItems – a collection of <InventoryItem> elements. Each <InventoryItem> can be an <AccommodationSegment> or a <TravelSegment>.

CustomerCounts – the number of customers in the relevant age categories (e.g. Adult, Child etc.)

TPA_Extensions – allows for bi-laterally-agreed additions to the base message

Sample Messages

The following sample messages address various ways to use the Package Availability Request message to establish the availability of a package holiday:

Example 1 – A standard package request

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_PkgAvailRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opentravel.org/OTA
OTA_PkgAvailRQ.xsd" TimeStamp="2001-04-08T21:10:30" EchoToken="7656" Target="Production" Version="1">
  <POS>
    <Source AgentSine="A1234">
      <UniqueId URL="www.anitetravel.com" Type="Profile" Id="a1234"/>
      <BookingChannel Type="Internet"/>
    </Source>
  </POS>
  <PackageRequest TourCode="ADABA" TravelCode="GAD20">
    <DateRange StartDate="2001-10-02" Duration="P14D"/>
    <CustomerCounts>
      <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
    </CustomerCounts>
  </PackageRequest>
</OTA_PkgAvailRQ>
```

```

        <CustomerCount AgeQualifyingCode="Child" Count="2"/>
        <CustomerCount AgeQualifyingCode="Infant" Count="1"/>
    </CustomerCounts>
</PackageRequest>
</OTA_PkgAvailRQ>

```

Example 2 – A one-way flight only request

```

<?xml version="1.0" encoding="UTF-8"?>
<OTA_PkgAvailRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opentravel.org/OTA
OTA_PkgAvailRQ.xsd" TimeStamp="2001-04-08T21:10:30" EchoToken="7656" Target="Production" Version="1">
    <POS>
        <Source AgentSine="A1234">
            <UniqueId URL="www.anitetravel.com" Type="Profile" Id="a1234"/>
            <BookingChannel Type="Internet"/>
        </Source>
    </POS>
    <PackageRequest PackageType="Flight Only" CompanyCode="COS">
        <DateRange StartDate="2001-10-02"/>
        <CustomerCounts>
            <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
        </CustomerCounts>
    </PackageRequest>
</OTA_PkgAvailRQ>

```

2.2.2. Package Availability Response - Success

The <OTA_PkgAvailRS> message returns a priced summary of the requested package or valid alternatives if any item of inventory cannot be satisfied. The components of a successful response are as follows:

- **OTA_v2ent** – indicates whether the original request was successful or unsuccessful and includes the standard attributes of the root tag of all OTA messages.

Package – provides complete details of the requested package. Its components are:

- **SKU** (Stock Keeping Unit) – the supplier's identification code for the full tour package
- **URL** (Uniform Resource Locator) – an alternative method of identifying the full tour package
- **ProductType** – identifies the type of product, e.g. Air Inclusive; Flight Only; Accommodation Only
- **AvailableQty** – the number of seats or beds remaining for sale.

DateRange – provides the start date and overall duration of the package

PackageSegments – A collection of <PackageSegment> elements making up a package itinerary.

Prices – a collection of package price elements and associated price rules

Extras – a collection of optional items which may be added to a basic package, e.g. excursions, car hire etc.

AcceptedPaymentForms - provides a list of the methods which may be used to pay for the holiday.

Package Segment

The components of <PackageSegment> are:

- **ItinerarySequence** – The position of this item in the package itinerary
- **ChronologicalSequence** - The position of this item in the package in time sequence. Note: it is possible for more than one inventory item to have the same ChronologicalSequence value e.g. Car Hire and Hotel Stay.

and any one of the following elements:

DestinationInformation - contains essential information regarding the resort or area of the chosen property which is appropriate to the enquirer's stay and could affect their decision to book. The components of <DestinationInformation> are:

- *ResortName* – describes the resort or area containing the requested property.
- Information** – defines the information text together with the period of applicability.

AccommodationInformation - summarises a property stay with associated facilities and essential information. The components of <AccommodationInformation> are:

- *RPH* (Reference Place Holder) – an index code to identify an instance in a collection of like items.
- *HotelName* – describes the requested property.
- *ResortName* – describes the resort or area of the requested property.
- DateRange** – defines the start date and duration of the property stay.
- RoomProfiles** – defines a set of room profiles identifying room types and occupancies.
- MealPlans** – defines the choice of board terms for the stay.
- Information** – defines the information text together with the period of applicability.

TransportInformation - summarises a travel segment and provides essential information when appropriate. The components of <TransportInformation> are:

- Travel** – defines a single travel leg.
- Information** – defines the information text together with the period of applicability.

Travel

The <Travel> element describes a travel arrangement which satisfies the customer's criteria. Only <AirSegment> is currently defined; other transport segments are planned e.g. <RailSegment>, <FerrySegment>.

The components of an <AirSegment> are:

- *RPH* (Reference Place Holder) – an index code to identify an instance in a collection of like items.
- *Type* – Identifies whether the segment is Charter or Scheduled.
- *TravelDateTimeAttributes*:
 - *PkgDepartureDay* – defines the day when the travel segment begins e.g. 'Monday' etc.
 - *PkgCheckIn* – defines check-in date and time.
 - *PkgDeparture* – defines the take-off date and time.
 - *PkgArrival* – defines the landing date and time.
- *TravelCode* – the supplier's code to identify the travel arrangement for the package.
- *Duration* – a length of time.
- *CarrierCode* – a code representing the transport operator.
- *CarrierName* - the name of the transport operator.
- *Serviceld* – the transport operator's code to identify a travel segment.
- DepartureAirport** – identifies the airport where the travel segment begins.
- ArrivalAirport** – identifies the airport where the travel segment ends.
- Supplement** – the supplement payable for this travel item.
- SeatAvailability** – a collection of seat availability by class.

Sample Message - success

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_PkgAvailRS xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA
OTA_PkgAvailRS.xsd" TimeStamp="2001-04-08T21:10:30" EchoToken="7656" Target="Production" Version="1">
  <Warnings>
    <Warning Type="BizRule">Please select room(s) and meal plan(s) from the accompanying lists</Warning>
  </Warnings>
```

```

<Package>
  <DateRange StartDate="2001-10-02" Duration="P14D"/>
  <PackageSegments>
    <PackageSegment>
      <TransportInformation>
        <Travel>
          <AirSegment RPH="1" PkgDepartureDay="Tue" PkgDeparture="2001-09-11T09:30:47-05:00"
PkgArrival="2001-09-11T09:30:47-05:00" TravelCode="GAD20">
            <DepartureAirport Code="LGW">London Gatwick</DepartureAirport>
            <ArrivalAirport Code="PMI">Palma</ArrivalAirport>
            <AvailableSeats>
              <SeatAvailability SeatQty="4"/>
            </AvailableSeats>
          </AirSegment>
        </Travel>
        <Information>N.B this flight is non-smoking</Information>
      </TransportInformation>
    </PackageSegment>
    <PackageSegment>
      <DestinationInformation ResortName="Alcudia">
        <Information FirstDate="2001-10-08" LastDate="2001-10-15">Please note that the October festival
in Alcudia is very noisy</Information>
      </DestinationInformation>
    </PackageSegment>
    <PackageSegment>
      <AccommodationInformation RPH="2" HotelName="Hotel Miramar" ResortName="Alcudia">
        <DateRange StartDate="2001-10-02" Duration="P14D"/>
        <RoomProfiles>
          <RoomProfile Code="TWIN" Description="Twin room with two extra beds and cot, bath, shower,
WC" Qty="1" CotQty="1">
            <CustomerCounts>
              <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
              <CustomerCount AgeQualifyingCode="Child" Count="2"/>
              <CustomerCount AgeQualifyingCode="Infant" Count="1"/>
            </CustomerCounts>
          </RoomProfile>
        </RoomProfiles>
        <MealPlans>
          <MealPlan>
            <BoardTerms Code="HB" Qty="4">Half Board</BoardTerms>
          </MealPlan>
        </MealPlans>
        <Information>The outdoor swimming pool will be closed for the winter from 14
October</Information>
      </AccommodationInformation>
    </PackageSegment>
    <PackageSegment>
      <TransportInformation>
        <Travel>
          <AirSegment RPH="3" PkgDepartureDay="Tue" PkgDeparture="2001-09-11T09:30:47-05:00"
PkgArrival="2001-09-11T09:30:47-05:00" TravelCode="GAD20">
            <DepartureAirport Code="PMI">Palma</DepartureAirport>
            <ArrivalAirport Code="LGW">London Gatwick</ArrivalAirport>
            <AvailableSeats>
              <SeatAvailability SeatQty="4"/>
            </AvailableSeats>
          </AirSegment>
        </Travel>
        <Information>N.B this flight is non-smoking</Information>
      </TransportInformation>
    </PackageSegment>
  </PackageSegments>
</Package>
<FacilityChoices>
  <AvailableRooms>
    <Room RPH="1" Code="SINGLE" Description="Single Room" AvailableQty="2" MinOccupancy="1"
MaxOccupancy="1">
      <Supplement PkgCurrencyCode="GBP">21.00</Supplement>
    </Room>
  </AvailableRooms>
</FacilityChoices>

```

```

    <Room RPH="2" Code="TWIN" Description=" Twin Room" AvailableQty="5" MinOccupancy="2"
MaxOccupancy="4" MaxAdults="3" CotQty="2" FreeChildFlag="true"/>
  </AvailableRooms>
  <RoomPrices BoardTerms="HB">
    <RoomPrice RPH="1" Code="SINGLE">
      <CustomerCounts>
        <CustomerCount AgeQualifyingCode="Adult" Count="1"/>
      </CustomerCounts>
      <ItemPrice Description="Adult Price">
        <UnitPrice>371.00</UnitPrice>
      </ItemPrice>
      <ProfilePrice>371.00</ProfilePrice>
    </RoomPrice>
    <RoomPrice RPH="2" Code="TWIN">
      <CustomerCounts>
        <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
      </CustomerCounts>
      <ItemPrice Description="Adult Price">
        <UnitPrice>350.00</UnitPrice>
      </ItemPrice>
      <ProfilePrice>700.00</ProfilePrice>
    </RoomPrice>
    <RoomPrice RPH="2" Code="TWIN">
      <CustomerCounts>
        <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
        <CustomerCount AgeQualifyingCode="Child" Count="1"/>
      </CustomerCounts>
      <ItemPrice Description="Adult Price" Qty="2">
        <UnitPrice>350.00</UnitPrice>
        <ExtendedPrice>700.00</ExtendedPrice>
      </ItemPrice>
      <ItemPrice Description="First Child">
        <UnitPrice>250.00</UnitPrice>
        <ExtendedPrice>250.00</ExtendedPrice>
      </ItemPrice>
      <ProfilePrice>950.00</ProfilePrice>
    </RoomPrice>
    <RoomPrice RPH="2" Code="TWIN">
      <CustomerCounts>
        <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
        <CustomerCount AgeQualifyingCode="Child" Count="2"/>
      </CustomerCounts>
      <ItemPrice Description="Adult Price" Qty="2">
        <UnitPrice>350.00</UnitPrice>
        <ExtendedPrice>700.00</ExtendedPrice>
      </ItemPrice>
      <ItemPrice Description="First Child">
        <UnitPrice>200.00</UnitPrice>
        <ExtendedPrice>200.00</ExtendedPrice>
      </ItemPrice>
      <ItemPrice Description="Second Child">
        <UnitPrice>250.00</UnitPrice>
        <ExtendedPrice>250.00</ExtendedPrice>
      </ItemPrice>
      <ProfilePrice>1150.00</ProfilePrice>
    </RoomPrice>
    <RoomPrice RPH="2" Code="TWIN">
      <CustomerCounts>
        <CustomerCount AgeQualifyingCode="Adult" Count="3"/>
      </CustomerCounts>
      <ItemPrice Description="Adult Price" Qty="3">
        <UnitPrice>308.00</UnitPrice>
      </ItemPrice>
      <ProfilePrice>924.00</ProfilePrice>
    </RoomPrice>
  </RoomPrices>
  <RoomPrices BoardTerms="FB">
    <RoomPrice RPH="1" Code="SINGLE">
      <CustomerCounts>
        <CustomerCount AgeQualifyingCode="Adult" Count="1"/>

```

```

        </CustomerCounts>
        <ItemPrice Description="Adult Price">
            <UnitPrice>611.00</UnitPrice>
        </ItemPrice>
        <ProfilePrice>611.00</ProfilePrice>
    </RoomPrice>
    <RoomPrice RPH="2" Code="TWIN">
        <CustomerCounts>
            <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
        </CustomerCounts>
        <ItemPrice Description="Adult Price" Qty="2">
            <UnitPrice>490.00</UnitPrice>
        </ItemPrice>
        <ProfilePrice>980.00</ProfilePrice>
    </RoomPrice>
    <RoomPrice RPH="2" Code="TWIN">
        <CustomerCounts>
            <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
            <CustomerCount AgeQualifyingCode="Child" Count="1"/>
        </CustomerCounts>
        <ItemPrice Description="Adult Price" Qty="2">
            <UnitPrice>490.00</UnitPrice>
            <ExtendedPrice>980.00</ExtendedPrice>
        </ItemPrice>
        <ItemPrice Description="First Child" Qty="1">
            <UnitPrice>390.00</UnitPrice>
            <ExtendedPrice>390.00</ExtendedPrice>
        </ItemPrice>
        <ProfilePrice>1370.00</ProfilePrice>
    </RoomPrice>
    <RoomPrice RPH="2" Code="TWIN">
        <CustomerCounts>
            <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
            <CustomerCount AgeQualifyingCode="Child" Count="2"/>
        </CustomerCounts>
        <ItemPrice Description="Adult Price" Qty="2">
            <UnitPrice>490.00</UnitPrice>
            <ExtendedPrice>980.00</ExtendedPrice>
        </ItemPrice>
        <ItemPrice Description="First Child">
            <UnitPrice>340.00</UnitPrice>
            <ExtendedPrice>340.00</ExtendedPrice>
        </ItemPrice>
        <ItemPrice Description="Second Child">
            <UnitPrice>390.00</UnitPrice>
            <ExtendedPrice>390.00</ExtendedPrice>
        </ItemPrice>
        <ProfilePrice>1710.00</ProfilePrice>
    </RoomPrice>
    <RoomPrice RPH="2" Code="TWIN">
        <CustomerCounts>
            <CustomerCount AgeQualifyingCode="Adult" Count="3"/>
        </CustomerCounts>
        <ItemPrice Description="Adult Price" Qty="3">
            <UnitPrice>448.00</UnitPrice>
        </ItemPrice>
        <ProfilePrice>1344</ProfilePrice>
    </RoomPrice>
</RoomPrices>
<AvailableMealPlans>
    <MealPlan>
        <BoardTerms Code="HB">Half Board</BoardTerms>
    </MealPlan>
    <MealPlan>
        <BoardTerms Code="FB">Full Board</BoardTerms>
        <Supplement PkgCurrencyCode="GBP" AgeQualifyingCode="Adult">140.00</Supplement>
        <Supplement PkgCurrencyCode="GBP" AgeQualifyingCode="Child">105.00</Supplement>
    </MealPlan>
</AvailableMealPlans>
</FacilityChoices>

```

</OTA_PkgAvailRS>

2.2.3. Package Availability Response - Warning

Where an item of Inventory cannot satisfy the request but alternatives are available, the <OTA_PkgAvailRS> message may return a warning together with details of these alternatives.

Components of an <OTA_PkgAvailRS> message with warnings are as follows:

- **OTA_v2ent** – indicates whether the original request was successful or unsuccessful and includes the standard attributes of the root tag of all OTA messages.

Package – describes the package holiday.

TravelChoices – a collection of <TravelItem> elements describing alternative travel arrangements, supplied when the original travel item cannot satisfy the request. The components of a <TravelItem> element are:

- **AccomOKFlag** – indicates whether or not the requested accommodation part of the package is available.

TravelDetail – provides details of travel arrangements by leg or:

TravelJourney – provides details of round-trip travel arrangements.

Supplements – a collection of <Supplement> elements defining additional charges over the base travel arrangement included in the package.

Information – defines the information text together with the period of applicability.

AccommodationChoices – a collection of <AvailableProperty> elements describing alternative accommodation items, supplied when the original accommodation item cannot satisfy the request. The components of an <AvailableProperty> element are:

- **SKU** (Stock Keeping Unit) – the supplier's identification code for the property.
- **URL** (Uniform Resource Locator) – an alternative method of identifying the full tour package
- **CheckInDateTime** - Date and (optionally) time of checkin (also known as Arrival Date/Time).
- **Duration** – the length of the stay.
- **CheckOutDateTime** - Normal Date and (optionally) time of checkout (also known as Departure Date/Time).
- **LastCheckOutDateTime** - Latest Date and (optionally) time of checkout.
- **MaxChildAge** – the maximum age to qualify for child prices or reductions.
- **TravelOKFlag** – indicates whether or not the requested travel part of the package is available.

Property – provides key details for the property.

Resort – identifies the resort containing the property

AvailableRooms – a collection of <Room> elements describing rooms available for booking.

AccommodationClass – identifies the national and supplier ratings for the property.

RoomPrices – a collection of <RoomPrice> elements setting out occupancy-level prices.

AvailableMealPlans – a collection of <MealPlan> elements describing meal arrangements available for booking.

SourceIdentification – indicates where details can be found, e.g. Brochure and Page Number.

FacilityChoices – a list of accommodation facilities from which the customer must choose in order for a successful request to be processed further. The components of

<FacilityChoices> are:

AvailableRooms – a collection of <Room> elements describing rooms available for booking.

RoomPrices – a collection of <RoomPrice> elements setting out occupancy-level prices.

AvailableMealPlans – a collection of <MealPlan> elements describing meal arrangements available for booking.

AvailableRooms

<AvailableRooms> comprises a multiple occurrence of <Room> to define the room arrangements available for the requested stay at the property. <Room> is a <RoomInventoryType>, which describes a type of room/cabin/apartment etc which is available for the requested stay at the property.

The components of <RoomInventoryType> are:

- *RPH* (Reference Place Holder) – an index code to identify an instance in a collection of like items
- *Code* – the code identifying this room type e.g. 'Single', 'Twin' etc.
- *Description* – a full description of the room (optional).
- *AvailableQty* – the quantity of this room type available for booking (optional).
- *MinOccupancy* – the minimum number of occupants for this room type (optional).
- *MaxOccupancy* – the maximum number of occupants for this room type (optional).
- *CotQty* – the number of cots which this room type can hold (optional).
- *FreeChildFlag* – indicates whether the room is available when the booking has a free child (optional).
- *RoomId* – identifies a specific room or accommodation unit (optional).

Supplement – the supplement payable for a room type.

RoomPrices

<RoomPrices> comprises a multiple occurrence of <RoomPrice> to define the prices for each available room type by occupancy profile. The components of <RoomPrice> used in this section are:

- *RPH* (Reference Place Holder) – an index code to identify the room type in the package element.
- *Code* – the Room Code.
- *Description* – a short description of the room type.

CustomerCounts – a collection of customer counts by age qualification e.g. Adults; Child.

ItemPrice – the unit price (e.g. per Adult) and the extended price (the unit price multiplied by the customer count for this age qualification) for each age qualification.

ProfilePrice – the sum of the extended prices for this room's occupancy profile.

AvailableMealPlans

<AvailableMealPlans> comprises a multiple occurrence of <MealPlan> to define the board arrangements available for the requested stay at the property

The components of <MealPlan> are:

- *BoardTerms* – the <BoardCode> identifying this meal plan e.g. "BB" (Bed & Breakfast), "HB" (Half Board) etc. and a <Qty> identifying the number of people taking the plan (if <CustomerCounts> is not necessary)

CustomerCounts – the number of customers by age category. Only necessary when the supplement is variable by age category.

Supplement – the supplement payable for this meal plan.

Sample Message – Warning

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_PkgAvailRS xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA
OTA_PkgAvailRS.xsd" TimeStamp="2001-04-08T21:10:30" EchoToken="7656" Target="Production" Version="1">
  <Warnings>
    <Warning Type="BizRule">Please select room(s) and meal plan(s) from the accompanying lists</Warning>
  </Warnings>
</Package>
```

```

<DateRange StartDate="2001-10-02" Duration="P14D"/>
<PackageSegments>
  <PackageSegment>
    <TransportInformation>
      <Travel>
        <AirSegment RPH="1" PkgDepartureDay="Tue" PkgDeparture="2001-09-11T09:30:47-05:00"
PkgArrival="2001-09-11T09:30:47-05:00" TravelCode="GAD20">
          <DepartureAirport Code="LGW">London Gatwick</DepartureAirport>
          <ArrivalAirport Code="PMI">Palma</ArrivalAirport>
          <AvailableSeats>
            <SeatAvailability SeatQty="4"/>
          </AvailableSeats>
        </AirSegment>
      </Travel>
      <Information>N.B this flight is non-smoking</Information>
    </TransportInformation>
  </PackageSegment>
  <PackageSegment>
    <DestinationInformation ResortName="Alcudia">
      <Information FirstDate="2001-10-08" LastDate="2001-10-15">Please note that the October festival
in Alcudia is very noisy</Information>
    </DestinationInformation>
  </PackageSegment>
  <PackageSegment>
    <AccommodationInformation RPH="2" HotelName="Hotel Miramar" ResortName="Alcudia">
      <DateRange StartDate="2001-10-02" Duration="P14D"/>
      <RoomProfiles>
        <RoomProfile Code="TWIN" Description="Twin room with two extra beds and cot, bath, shower,
WC" Qty="1" CotQty="1">
          <CustomerCounts>
            <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
            <CustomerCount AgeQualifyingCode="Child" Count="2"/>
            <CustomerCount AgeQualifyingCode="Infant" Count="1"/>
          </CustomerCounts>
        </RoomProfile>
      </RoomProfiles>
      <MealPlans>
        <MealPlan>
          <BoardTerms Code="HB" Qty="4">Half Board</BoardTerms>
        </MealPlan>
      </MealPlans>
      <Information>The outdoor swimming pool will be closed for the winter from 14
October</Information>
    </AccommodationInformation>
  </PackageSegment>
  <PackageSegment>
    <TransportInformation>
      <Travel>
        <AirSegment RPH="3" PkgDepartureDay="Tue" PkgDeparture="2001-09-11T09:30:47-05:00"
PkgArrival="2001-09-11T09:30:47-05:00" TravelCode="GAD20">
          <DepartureAirport Code="PMI">Palma</DepartureAirport>
          <ArrivalAirport Code="LGW">London Gatwick</ArrivalAirport>
          <AvailableSeats>
            <SeatAvailability SeatQty="4"/>
          </AvailableSeats>
        </AirSegment>
      </Travel>
      <Information>N.B this flight is non-smoking</Information>
    </TransportInformation>
  </PackageSegment>
</PackageSegments>
</Package>
<FacilityChoices>
  <AvailableRooms>
    <Room RPH="1" Code="SINGLE" Description="Single Room" AvailableQty="2" MinOccupancy="1"
MaxOccupancy="1">
      <Supplement PkgCurrencyCode="GBP">21.00</Supplement>
    </Room>
    <Room RPH="2" Code="TWIN" Description="Twin Room" AvailableQty="5" MinOccupancy="2"
MaxOccupancy="4" MaxAdults="3" CotQty="2" FreeChildFlag="true"/>

```

```

</AvailableRooms>
<RoomPrices BoardTerms="HB">
  <RoomPrice RPH="1" Code="SINGLE">
    <CustomerCounts>
      <CustomerCount AgeQualifyingCode="Adult" Count="1"/>
    </CustomerCounts>
    <ItemPrice Description="Adult Price">
      <UnitPrice>371.00</UnitPrice>
    </ItemPrice>
    <ProfilePrice>371.00</ProfilePrice>
  </RoomPrice>
  <RoomPrice RPH="2" Code="TWIN">
    <CustomerCounts>
      <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
    </CustomerCounts>
    <ItemPrice Description="Adult Price">
      <UnitPrice>350.00</UnitPrice>
    </ItemPrice>
    <ProfilePrice>700.00</ProfilePrice>
  </RoomPrice>
  <RoomPrice RPH="2" Code="TWIN">
    <CustomerCounts>
      <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
      <CustomerCount AgeQualifyingCode="Child" Count="1"/>
    </CustomerCounts>
    <ItemPrice Description="Adult Price" Qty="2">
      <UnitPrice>350.00</UnitPrice>
      <ExtendedPrice>700.00</ExtendedPrice>
    </ItemPrice>
    <ItemPrice Description="First Child">
      <UnitPrice>250.00</UnitPrice>
      <ExtendedPrice>250.00</ExtendedPrice>
    </ItemPrice>
    <ProfilePrice>950.00</ProfilePrice>
  </RoomPrice>
  <RoomPrice RPH="2" Code="TWIN">
    <CustomerCounts>
      <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
      <CustomerCount AgeQualifyingCode="Child" Count="2"/>
    </CustomerCounts>
    <ItemPrice Description="Adult Price" Qty="2">
      <UnitPrice>350.00</UnitPrice>
      <ExtendedPrice>700.00</ExtendedPrice>
    </ItemPrice>
    <ItemPrice Description="First Child">
      <UnitPrice>200.00</UnitPrice>
      <ExtendedPrice>200.00</ExtendedPrice>
    </ItemPrice>
    <ItemPrice Description="Second Child">
      <UnitPrice>250.00</UnitPrice>
      <ExtendedPrice>250.00</ExtendedPrice>
    </ItemPrice>
    <ProfilePrice>1150.00</ProfilePrice>
  </RoomPrice>
  <RoomPrice RPH="2" Code="TWIN">
    <CustomerCounts>
      <CustomerCount AgeQualifyingCode="Adult" Count="3"/>
    </CustomerCounts>
    <ItemPrice Description="Adult Price" Qty="3">
      <UnitPrice>308.00</UnitPrice>
    </ItemPrice>
    <ProfilePrice>924.00</ProfilePrice>
  </RoomPrice>
</RoomPrices>
<RoomPrices BoardTerms="FB">
  <RoomPrice RPH="1" Code="SINGLE">
    <CustomerCounts>
      <CustomerCount AgeQualifyingCode="Adult" Count="1"/>
    </CustomerCounts>
    <ItemPrice Description="Adult Price">

```

```

        <UnitPrice>611.00</UnitPrice>
    </ItemPrice>
    <ProfilePrice>611.00</ProfilePrice>
</RoomPrice>
<RoomPrice RPH="2" Code="TWIN">
    <CustomerCounts>
        <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
    </CustomerCounts>
    <ItemPrice Description="Adult Price" Qty="2">
        <UnitPrice>490.00</UnitPrice>
    </ItemPrice>
    <ProfilePrice>980.00</ProfilePrice>
</RoomPrice>
<RoomPrice RPH="2" Code="TWIN">
    <CustomerCounts>
        <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
        <CustomerCount AgeQualifyingCode="Child" Count="1"/>
    </CustomerCounts>
    <ItemPrice Description="Adult Price" Qty="2">
        <UnitPrice>490.00</UnitPrice>
        <ExtendedPrice>980.00</ExtendedPrice>
    </ItemPrice>
    <ItemPrice Description="First Child" Qty="1">
        <UnitPrice>390.00</UnitPrice>
        <ExtendedPrice>390.00</ExtendedPrice>
    </ItemPrice>
    <ProfilePrice>1370.00</ProfilePrice>
</RoomPrice>
<RoomPrice RPH="2" Code="TWIN">
    <CustomerCounts>
        <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
        <CustomerCount AgeQualifyingCode="Child" Count="2"/>
    </CustomerCounts>
    <ItemPrice Description="Adult Price" Qty="2">
        <UnitPrice>490.00</UnitPrice>
        <ExtendedPrice>980.00</ExtendedPrice>
    </ItemPrice>
    <ItemPrice Description="First Child">
        <UnitPrice>340.00</UnitPrice>
        <ExtendedPrice>340.00</ExtendedPrice>
    </ItemPrice>
    <ItemPrice Description="Second Child">
        <UnitPrice>390.00</UnitPrice>
        <ExtendedPrice>390.00</ExtendedPrice>
    </ItemPrice>
    <ProfilePrice>1710.00</ProfilePrice>
</RoomPrice>
<RoomPrice RPH="2" Code="TWIN">
    <CustomerCounts>
        <CustomerCount AgeQualifyingCode="Adult" Count="3"/>
    </CustomerCounts>
    <ItemPrice Description="Adult Price" Qty="3">
        <UnitPrice>448.00</UnitPrice>
    </ItemPrice>
    <ProfilePrice>1344</ProfilePrice>
</RoomPrice>
</RoomPrices>
<AvailableMealPlans>
    <MealPlan>
        <BoardTerms Code="HB">Half Board</BoardTerms>
    </MealPlan>
    <MealPlan>
        <BoardTerms Code="FB">Full Board</BoardTerms>
        <Supplement PkgCurrencyCode="GBP" AgeQualifyingCode="Adult">140.00</Supplement>
        <Supplement PkgCurrencyCode="GBP" AgeQualifyingCode="Child">105.00</Supplement>
    </MealPlan>
</AvailableMealPlans>
</FacilityChoices>
</OTA_PkgAvailRS>

```

2.3. Create Booking

The Create Booking messages are designed to make a confirmed booking of a package holiday whose availability may or may not have been checked. An <ActionType> qualifier is available to modify the default 'Book' request to simply return a Quotation or make a provisional reservation pending authorisation of payment details.

If the 'Book' action request is satisfied, the enquirer will be requested to provide contact and payment details. On authorisation of the payment details the enquirer will be provided with a Booking Reference (and, optionally, Invoice details for printing).

2.4. Create Booking Use Case

The business use case that supports this message identifies a customer or agent (person or system acting on behalf of the customer) who requires to book a specific occurrence of a package. The first step in this use case is for the traveller or requesting party to supply the package and party details.

The steps of the use case proceed as follows:

1. The Customer or agent requests the creation of a booking for a specific package for a date and duration for a number of passengers, together with contact details.
2. The system reserves the necessary capacity and confirms the details of the booking including costs.
3. When necessary the customer or agent provides payment details and the supplier obtains payment authorisation.
4. The supplier creates a booking entity and provides the customer or agent with the Booking Reference and optionally the data to produce a written confirmation.

Possible business processing errors include:

- One or more elements of the package cannot satisfy the number of passengers for the date and duration requested in which circumstance the use case will revert to the Package Availability response message as described in the Package Availability scenario in section 1 of this document.
- Payment authorisation is refused.

Scenario

A customer wishes to book the package consisting of the Hotel Miramar in Alcudia Majorca travelling on a specific return flight pair between London Gatwick and Palma for 2 adults and 2 children and one infant for 14 nights from 02 October 2001.

2.5. Create Booking Messages

The OTA version 2001B specification of Create Booking provides a set of messages to support the functionality of creating a package booking.

The syntax of the action verbs that are the root elements of a payload document used to create a booking are as follows:

- **OTA_PkgBookRQ** – Provides the identification criteria for the package components, passenger and contact details.
- **OTA_PkgBookRS** – Provides complete details of the booking with costs and methods of payment.

2.5.1. Create Booking Request

The OTA_PkgBookRQ message identifies the required package and supplies passenger details for the booking.

The root element of the <OTA_PkgBookRQ> contains the standard payload attributes found in all OTA payload documents as well as the attribute <ReqRespVersion> that requests a specific version of the response message. As this is the first publication of the OTA package availability message set, currently the only valid value is "1".

The other components are:

- *ActionType* identifies the action to be taken – Book, Hold, Quote or Confirm.

POS – identifies the requestor.

Uniqueld – identifies a provisional reservation being retrieved for confirmation.

PackageRequest – identifies the package to be satisfied and defines the date, duration and customers by age category.

ContactDetail – Details of the person to whom correspondence regarding a booking will be directed.

PassengerListItems – a collection of <PassengerListItem> elements each providing identity details of a single passenger. The components of a <PassengerListItem> are:

- *PassengerRPH* – the index used to identify a passenger.
- *Gender* – A code indicating the gender of a passenger.
- *Nationality* – A code to identify the country of origin of a person by birth or naturalisation.
- *Age* – the age of a passenger, usually required as at the end of the stay.
- *BirthDate* – the date of birth of a passenger.

PersonName – identifies an individual passenger by name.

SpecialNeeds – a collection of <SpecialNeed> items.

PersonalInsuranceItems – a collection of <PersonalInsuranceItem> elements which identifies the type of personal insurance and number of customers covered. The components of a <PersonalInsuranceItem> are:

- *Code* – A coded identification of a type of insurance.

CustomerCounts – the number of customers in the relevant age categories (e.g. Adult, Child etc.).

OwnInsuranceChoice – a collection of <OwnInsuranceChoice> elements providing details of the customer's own insurance policy which is being used in place of the supplier's offering. The components of <OwnInsuranceChoice> are:

- *InsuranceCompany* – the name of the customer's chosen insurance company.
- *PolicyNumber* – a reference by which an issued policy is recognised.

CustomerCounts – the number of customers in the relevant age categories (e.g. Adult, Child etc.).

PaymentDetails – a collection of <PaymentDetail> elements which define a monetary amount and a method of payment. This will commonly be the only element, apart from the Uniqueld, supplied for a 'Confirm' action. The components of <PaymentDetail> are:

PaymentForm – defines the method of payment. See OTA_Profile for description.

Amount – the monetary value to be paid with the method of payment.

Sample Message – Book

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_PkgBookRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA
OTA_PkgBookRQ.xsd" TimeStamp="2001-04-08T21:10:30" EchoToken="7656" Target="Production" Version="1"
ActionType="Book">
  <Uniqueld Type="Reservation" Id="1234566"/>
  <PackageRequest PackageType="Flight Only" CompanyCode="ABC" TourCode="DEF" TravelCode="VIEW"
BoardCode="AB" SKU="20022001" URL="http://www.mytravelpartner.com" PromotionCode="AB-SELL">
    <InventoryItems>
```

```

<InventoryItem>
  <Travel>
    <AirSegment RPH="1" PkgDepartureDay="Tue" PkgDeparture="2001-09-11T09:30:47-05:00"
PkgArrival="2001-09-11T09:30:47-05:00" TravelCode="GAD20">
      <AvailableSeats>
        <SeatAvailability SeatQty="4"/>
      </AvailableSeats>
    </AirSegment>
  </Travel>
</InventoryItem>
<InventoryItem>
  <Accommodation RPH="2">
    <HotelReference HotelCode="ADABA"/>
    <DateRange StartDate="2001-10-02" Duration="P14D"/>
    <RoomProfiles>
      <RoomProfile RPH="01" Code="TWIN" Qty="1" CotQty="1">
        <CustomerCounts>
          <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
          <CustomerCount AgeQualifyingCode="Child" Count="2"/>
          <CustomerCount AgeQualifyingCode="Infant" Count="1"/>
        </CustomerCounts>
        <PassengerRPHs>
          <PassengerRPH RPH="1"/>
          <PassengerRPH RPH="2"/>
          <PassengerRPH RPH="3"/>
          <PassengerRPH RPH="4"/>
          <PassengerRPH RPH="5"/>
        </PassengerRPHs>
      </RoomProfile>
    </RoomProfiles>
    <MealPlans>
      <MealPlan>
        <BoardTerms Code="HB" Qty="4"/>
      </MealPlan>
    </MealPlans>
  </Accommodation>
</InventoryItem>
<InventoryItem>
  <Travel>
    <AirSegment RPH="3" PkgDepartureDay="Tue" PkgDeparture="2001-09-11T09:30:47-05:00"
PkgArrival="2001-09-11T09:30:47-05:00" TravelCode="GAD20">
      <AvailableSeats>
        <SeatAvailability SeatQty="4"/>
      </AvailableSeats>
    </AirSegment>
  </Travel>
</InventoryItem>
</InventoryItems>
</PackageRequest>
<ContactDetail>
  <Name>
    <NamePrefix>Mr</NamePrefix>
    <GivenName>S</GivenName>
    <Surname>Corcoran</Surname>
    <PrivacyDetails/>
  </Name>
</ContactDetail>
<PassengerListItems>
  <PassengerListItem RPH="01" Gender="M" Nationality="GB" BirthDate="1972-02-27">
    <Name>
      <NamePrefix>Mr</NamePrefix>
      <GivenName>Simon</GivenName>
      <Surname>Corcoran</Surname>
      <PrivacyDetails/>
    </Name>
  </PassengerListItem>
  <PassengerListItem RPH="02" Gender="F" Nationality="US" Age="24">
    <Name>
      <NamePrefix>Mrs</NamePrefix>
      <GivenName>A</GivenName>

```

```

        <Surname>Corcoran</Surname>
        <PrivacyDetails/>
    </Name>
</PassengerListItem>
<PassengerListItem RPH="03" Gender="F" Nationality="GB" Age="8">
    <Name>
        <NamePrefix>Miss</NamePrefix>
        <GivenName>B</GivenName>
        <Surname>Corcoran</Surname>
        <PrivacyDetails/>
    </Name>
</PassengerListItem>
<PassengerListItem RPH="04" Gender="M" Nationality="GB" Age="4">
    <Name>
        <NamePrefix>Mstr</NamePrefix>
        <GivenName>C</GivenName>
        <Surname>Corcoran</Surname>
        <PrivacyDetails/>
    </Name>
</PassengerListItem>
<PassengerListItem RPH="05" Gender="F" Nationality="GB" Age="1">
    <Name>
        <NamePrefix>Inf</NamePrefix>
        <GivenName>D</GivenName>
        <Surname>Corcoran</Surname>
        <PrivacyDetails/>
    </Name>
</PassengerListItem>
</PassengerListItems>
<OwnInsuranceChoices>
    <OwnInsuranceChoice InsuranceCompany="Home and Colonial" PolicyNmbr="1453891762">
        <CustomerCounts>
            <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
            <CustomerCount AgeQualifyingCode="Child" Count="2"/>
            <CustomerCount AgeQualifyingCode="Infant" Count="1"/>
        </CustomerCounts>
    </OwnInsuranceChoice>
</OwnInsuranceChoices>
</OTA_PkgBookRQ>

```

Sample Message – Confirm

```

<?xml version="1.0" encoding="UTF-8"?>
<OTA_PkgBookRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA
OTA_PkgBookRQ.xsd" TimeStamp="2001-04-08T21:10:30" EchoToken="7656" Target="Production" Version="1"
ActionType="Confirm">
    <UniqueId Type="Reservation" Id="AA12345"/>
    <PaymentDetails>
        <PaymentDetail>
            <CreditCard CardType="Credit" CardNumber="4444 1234 5678 5532" EffectiveDate="2004-08-13"
ExpireDate="2004-08-13">
                <CardHolderName>
                    <NamePrefix>Mr</NamePrefix>
                    <GivenName>S</GivenName>
                    <Surname>Corcoran</Surname>
                    <PrivacyDetails/>
                </CardHolderName>
                <CardIssuerName IssuingBankId="VISA"/>
                <Address>
                    <AddressLine>
                        <AddressLine>10 This Road</AddressLine>
                        <AddressLine>Holborn</AddressLine>
                    </AddressLine>
                    <CityName PostalCode="EC4 1HJ">London</CityName>
                    <CountryName CountryCode="UK">United Kingdom</CountryName>
                    <PrivacyDetails/>
                </Address>
            </CreditCard>
        </PaymentDetail>
    </PaymentDetails>

```

```

        <PrivacyDetails/>
    </CreditCard>
    <PrivacyDetails/>
    <Amount PkgCurrencyCode="GBP">1456.00</Amount>
</PaymentDetail>
</PaymentDetails>
</OTA_PkgBookRQ>

```

2.5.2. Create Booking Response

The OTA_PkgBookRS message provides full details of the requested booking.

The components of OTA_PkgBookRS are as follows:

- **OTA_v2ent** – indicates whether the original request was successful or unsuccessful and includes the standard attributes of the root tag of all OTA messages.

PackageReservation – details the booking as recorded by the supplier. The components of <PackageReservation> are:

POS – identifies the requestor.

Uniqueld – identifies a provisional reservation being retrieved for confirmation.

InventoryItems – a collection of <InventoryItem> elements. Each <InventoryItem> can be an <AccommodationSegment> or a <TravelSegment>.

ContactDetail – Details of the person to whom correspondence regarding a booking will be directed.

PassengerListItems – a collection of <PassengerListItem> elements.

PersonalInsuranceItems – a collection of <PersonalInsuranceItem> elements.

OwnInsuranceChoice – a collection of <OwnInsuranceChoice> elements.

InvoiceDetail – a list of all the items making up a priced booking.

InformationItems – a collection of the information required for individual inventory items.

Invoice Detail

<InvoiceDetail> is a list of all the items making up a priced booking. The components of <InvoiceDetail> are:

- **BalanceDueDate** – the date on which the BalanceAmount is payable by the customer.
- CostingItems** – is a collection of <CostingItem> elements which define an individually priced component of a priced booking. The components of <CostingItem> are:

- **Description** – a description of the priced item.

- **Quantity** – the number of units required.

UnitPrice – the cost of one unit of the component.

ExtendedPrice – unit price times quantity..

GrossAmount – defines the total amount payable by the customer.

DepositAmount – defines the amount payable at the time of booking to secure the booking.

DiscountCommission – defines the fee earned by the agent for the booking.

TaxItems – defines the tax amount(s) payable on an invoice.

NetAmount – the amount payable by the agent, i.e. GrossAmount less Discount/Commission (optional).

BalanceDueAmount – the amount remaining to be paid by the customer i.e. GrossAmount less DepositAmount.

Sample message

```

<?xml version="1.0" encoding="UTF-8"?>
<OTA_PkgBookRS xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA
OTA_PkgBookRS.xsd" EchoToken="7656" TimeStamp="2001-04-08T21:10:30" Target="Production" Version="1"
SequenceNbr="3">
    <Success/>

```

```

<PackageReservation>
  <UniqueId Type="Reservation" Id="AA12345"/>
  <InventoryItems>
    <InventoryItem ChronologicalSequence="1" ItinerarySequence="1">
      <Travel>
        <AirSegment RPH="1" PkgDepartureDay="Tue" PkgDeparture="2001-09-11T09:30:47-05:00"
PkgArrival="2001-09-11T09:30:47-05:00" TravelCode="GAD20">
          <DepartureAirport Code="LGW">London Gatwick</DepartureAirport>
          <ArrivalAirport Code="PMI">Palma</ArrivalAirport>
          <AvailableSeats>
            <SeatAvailability SeatQty="4"/>
          </AvailableSeats>
        </AirSegment>
      </Travel>
    </InventoryItem>
    <InventoryItem ChronologicalSequence="2" ItinerarySequence="2">
      <Accommodation RPH="2">
        <HotelReference HotelCode="ADAB"/>
        <DateRange StartDate="2001-10-02" Duration="P14D"/>
        <RoomProfiles>
          <RoomProfile Code="TWIN" Description="Twin room with two extra beds and cot, bath, shower,
WC" Qty="1" CotQty="1">
            <CustomerCounts>
              <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
              <CustomerCount AgeQualifyingCode="Child" Count="2"/>
              <CustomerCount AgeQualifyingCode="Infant" Count="1"/>
            </CustomerCounts>
            </RoomProfile>
          </RoomProfiles>
          <MealPlans>
            <MealPlan>
              <BoardTerms Code="HB" Qty="4">Half Board</BoardTerms>
            </MealPlan>
          </MealPlans>
        </Accommodation>
      </InventoryItem>
      <InventoryItem ChronologicalSequence="3" ItinerarySequence="3">
        <Travel>
          <AirSegment RPH="3" PkgDepartureDay="Tue" PkgDeparture="2001-09-11T09:30:47-05:00"
PkgArrival="2001-09-11T09:30:47-05:00" TravelCode="GAD20">
            <DepartureAirport Code="PMI">Palma</DepartureAirport>
            <ArrivalAirport Code="LGW">London Gatwick</ArrivalAirport>
            <AvailableSeats>
              <SeatAvailability SeatQty="4"/>
            </AvailableSeats>
          </AirSegment>
        </Travel>
      </InventoryItem>
    </InventoryItems>
    <ContactDetail>
      <Name>
        <NamePrefix>Mr</NamePrefix>
        <GivenName>S</GivenName>
        <Surname>Corcoran</Surname>
        <PrivacyDetails/>
      </Name>
    </ContactDetail>
    <PassengerListItems>
      <PassengerListItem RPH="01" Gender="M" Nationality="GB" BirthDate="1972-02-27">
        <Name>
          <NamePrefix>Mr</NamePrefix>
          <GivenName>Simon</GivenName>
          <Surname>Corcoran</Surname>
          <PrivacyDetails/>
        </Name>
      </PassengerListItem>
      <PassengerListItem RPH="02" Gender="F" Nationality="US" Age="24">
        <Name>
          <NamePrefix>Mrs</NamePrefix>
          <GivenName>A</GivenName>

```

```

        <Surname>Corcoran</Surname>
        <PrivacyDetails/>
    </Name>
</PassengerListItem>
<PassengerListItem RPH="03" Gender="F" Nationality="GB" Age="8">
    <Name>
        <NamePrefix>Miss</NamePrefix>
        <GivenName>B</GivenName>
        <Surname>Corcoran</Surname>
        <PrivacyDetails/>
    </Name>
</PassengerListItem>
<PassengerListItem RPH="04" Gender="M" Nationality="GB" Age="4">
    <Name>
        <NamePrefix>Mstr</NamePrefix>
        <GivenName>C</GivenName>
        <Surname>Corcoran</Surname>
        <PrivacyDetails/>
    </Name>
</PassengerListItem>
<PassengerListItem RPH="05" Gender="F" Nationality="GB" Age="1">
    <Name>
        <NamePrefix>Inf</NamePrefix>
        <GivenName>D</GivenName>
        <Surname>Corcoran</Surname>
        <PrivacyDetails/>
    </Name>
</PassengerListItem>
</PassengerListItems>
<InvoiceDetail>
    <CostingItems>
        <CostingItem Description="Basic Holiday Price" Qty="2">
            <UnitPrice>546.00</UnitPrice>
            <ExtendedPrice>1092.00</ExtendedPrice>
        </CostingItem>
        <CostingItem Description="First Child" Qty="1">
            <UnitPrice>150.00</UnitPrice>
            <ExtendedPrice>150.00</ExtendedPrice>
        </CostingItem>
        <CostingItem Description="Second Child" Qty="1">
            <UnitPrice>223.00</UnitPrice>
            <ExtendedPrice>223.00</ExtendedPrice>
        </CostingItem>
    </CostingItems>
    <GrossAmount>1465.00</GrossAmount>
    <DepositAmount>300.00</DepositAmount>
    <BalanceDueAmount>1165.00</BalanceDueAmount>
</InvoiceDetail>
</PackageReservation>
<InformationItems>
    <InformationItem InventorySequence="2" Type="Resort">
        <Information FirstDate="2001-10-02" Duration="P14D">Please note that the October festival in Alcudia is
very noisy</Information>
    </InformationItem>
</InformationItems>
</OTA_PkgBookRS>

```

Section 3. – Golf Tee Times

The OTA version 2001C specification of Golf Tee Times provides three separate request/response pairs of messages to support the functionality of requesting data from another system in the process of finding a golf course, inquiring as to availability, and booking a tee time. All message sets assumes a pull model, where the originating system requests a specific set of data (as agreed by trading partners).

A system through which an interested party initiates a booking process will request a list of courses that meet the specified qualifiers using the OTA_CourseSearchRQ message. In this message, the desired criteria date is identified. The receiving system (typically a Golf Course Tee Sheet System or a consolidator's system) responds with the OTA_CourseSearchRS message, which includes either summary or detailed information about the courses that meet the requested criteria. Where the flag DetailResponse is set to "Yes", all traits of the course(s) meeting the criteria are returned. Where the flag is set to "No", only those traits matching the requested criteria and the basic course information are returned. All messages assume the no-state, meaning that the querying system will initiate the transaction and expect a response from the queried system. All message responses include the request identification. Responses may be returned in any order.

OTA_CourseSearchRQ – Sends a request for course information to another system. All the elements and attributes are optional, unless otherwise stated as required. The requesting system may request a detailed or summary response.

OTA_CourseSearchRS - Returns a set of data representing the course(s) that meet the requested criteria. Where the criteria attribute of Required is "Yes" then only those courses that meet those criteria will be returned. Where the Required attribute is "No" then a course that does not meet that criteria may be included in the set. In all cases, where the criteria has been included in the request, the comparable trait and its value will be returned, along with the basic course information and identification. The message may also include Warnings from business processing rules or Errors if the request did not succeed.

OTA_CourseAvailRQ – Sends a request for course availability to another system. All the elements and attributes are optional, unless otherwise stated as required.

OTA_CourseAvailRS - Returns the requested set of data if the request can be processed, or includes Warnings from business processing rules or Errors if the request did not succeed.

OTA_CourseResRQ – Sends a request for a reservation to another system. All the elements and attributes are optional, unless otherwise stated as required.

OTA_CourseResRS - Returns the requested reservation if the request can be processed, or includes Warnings from business processing rules or Errors if the request did not succeed.

Note:

All the elements are indicated in **Bold**.

All the attributes are indicated in *italics*.

Example:

Element – Element Description

- *Attribute* – Attribute Description

3.1. Golf Course Search Request Message (OTA_CourseSearchRQ)

The root tag of OTA_CourseSearchRQ contains the standard payload attributeGroup OTA_PayloadStdAttributes, found in all OTA payload documents.

OTA_CourseSearchRQ - This is the main Request element.

- *OTA_PayloadStdAttributes*- includes standard attributes of the root tag of all OTA messages as defined in the OTA Version 2 Infrastructure Document.
- *GolfCourseID* - If a GolfCourseID is included in the request, then the response will be for only that golf course. If no GolfCourseID is included, the response will include all golf courses that are deemed to meet the requested criteria.
- *DetailResponse* - If the requestor desires a response that includes all the traits associated with the golf course(s) in the response, this boolean is set to "Yes". In that case, all the traits of the course will be returned. Where the DetailResponse is set to "No", the response will be required to send only the traits that match the criteria being sent.

Criteria - Criteria is a repeating set of features that are desired in the search for a golf course. The Name and Value pair define the criteria for the search. If the requestor demands that the result be filtered on a particular criterion, then the Required boolean is set to "Yes". If the Required boolean is not set to "Yes", then the response can include courses that do not meet those exact criteria. Examples would be:

Name="Architect", Value="Robert Trent Jones", Required="Yes".

Name="Location", Value="Myrtle Beach", Required="Yes".

Name="Caddies", Value="Yes", Required="No".

Name="Length", Value="6600 Yds", Required="Yes", Operation=">".

The Name, Value, and Required attributes are required, but the Operation is

optional.

- *Name* - A code representing the specific Criterion on which the search is to filter.
- *Value* - The value of the Criterion.
- *Required* - A flag establishing if this criterion must be met in a successful candidate. If "Yes", then all responses must meet the specified criterion.
- *Operation* - Where a test against a criterion value is not an equality, this represents the operation to be used as the filter. Examples would be GT (Greater Than), LT (Less Than), etc

3.2. Golf Course Search Response Message (OTA_CourseSearchRS)

The root tag of <OTA_CourseSearchRS> uses the external *include* <OTA_v2ent.xsd> that defines the root element standard attributes found in all the OTA payload documents, and the response options of returning the indication of Success, Warning or Errors in processing the request. The response message may include Warnings from business processing rules or Errors if the request did not succeed.

OTA_CourseSearchRS - Main Response element.

- *OTA_PayloadStdAttributes* - See above

Success

Warnings

Warning

- *Type* - Type of warning. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing)
- *Code* - The code assigned to the warnin
- *DocURL*>
- *Status*
- *Tag*

Errors

Error

- *Type* - Type of error. Valid values: (Unknown |NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing)
- *Code* - The code assigned to the warning
- *DocURL*
- *Status*
- *Tag*

- *DetailResponse* – See OTA_CourseSearchRQ

GolfCourse - Specific information about the golf course(s) for which the response is being made.

- *Id* - The Unique Identifier associated with the golf course
- *Name* - The name of the golf course
- *City* - The city in which the golf course is located
- *State* - The state in which the golf course is located
- *Country* - The country in which the golf course is located
- *PostalCode* - The postal code of the golf course
- *County* - The county in which the golf course is located
- *Province* - Where applicable, the province in which the golf course is located
- *PhoneNumber* - The phone number of the golf course

Trait - A repeating set of features that describe the golf course. A Trait consists of a Name and Value pair. Examples would be:

Name="Architect", Value="Robert Trent Jones".
Name="Location", Value="Myrtle Beach".
Name="Caddies", Value="Yes".
Name="Length", Value="6600 Yds"

Where the request (OTA_CourseSearchRQ) has the attribute value of DetailResponse="No", all the requested criteria should be returned as Traits with their associated value. If the criteria is not designated as Required="Yes", that criteria need not be met, but the value associated with that trait should be returned. Where the DetailResponse="Yes" then all traits of the golf course should be returned with their associated value.

- *Name* - The code representing a specific trait
- *Value* - The value associated with that trait

3.3. Course Availability Request Message (OTA_CourseAvailRQ)

The root tag of OTA_StatisticsRQ uses the external *include* <OTA_v2ent.xsd> that defines the root element standard attributes found in all the OTA payload documents

OTA_CourseAvailRQ - The OTA_CourseAvailRQ message is used to request availability at a known single course for one or more potential tee times. The specific information about the golfer or golfers is necessary in order to validate booking rules and set rates. It is the primary request element.

- *OTA_PayloadStdAttributes* - See above
- *GolfCourseID* - A unique identifier assigned to the golf course

GolfCourseTeeTime - The GolfCourseTeeTime attributes include the range of dates and times for which the round (or rounds) are being requested, as well as information relating to the number of golfers, the number of holes requested, and the number of times. It also includes the maximum price permitted in the response

- *StartDate* - The first date for which the availability request is valid
- *EndDate* - The last date for which the availability request is valid. If the request is for one day only, this will be the same as the StartDate
- *StartTime* - The earliest time for which the availability request is valid
- *EndTime* - The latest start time for which the availability request is valid. The StartTime and EndTime refer to each day of the period represented by the StartDate and EndDate. Where the start and end time varies by day, separate requests must be sent
- *NumberOfGolfers* - The number of golfers covered by this request.
- *NumberOfHoles* - The number of holes that the golfers wish to play within one round. If this differs by day, separate messages must be sent.
- *NumberOfTimes* - The number of Tee Times covered by this request. For example, if a foursome wants to play two separate rounds within one day, the number of tee times would be 2
- *MaxPrice* - The highest price acceptable for the requested rounds. Where a number of courses are represented by a system, only those rounds that are at or below the highest price should be returned
- *Currency* - The ISO currency code in which the price is to be quoted, and in which the max price is stated

RateQualifier - The RateQualifier is a code or set of repetitive codes that represent a business agreement between trading partners, or a request for a rate that might be associated with the requesting party. For example, if the requestor is a wholesaler or reseller, the RateQualifier will be a code associated with that status. If the individual golfer is a member of an organization that typically gets a discount (such as NGCOA, PGA, USGA, etc.) then that code would be included in the rate qualifier.

Memberships - Memberships are associated with one or more of the golfers for whom the specific request for availability is being made. Some memberships may permit booking rules that do not apply to other golfers. For example, a reciprocal membership may allow booking farther in advance than the standard golfer is permitted. NOTE: a definition of Memberships should be reflected in the OTA_Profile schema.

Membership - An individual membership.

- *Type* - The type of membership, using a code to represent the specific entity. Examples would be PGA, NGCOA, USGA, etc.
- *ID* - The Identification of the party within the stated membership

3.4. Course Availability Response Message (OTA_CourseAvailRS)

The root tag of OTA_StatisticsNotifRS uses the external *include* OTA_v2ent.xsd that defines the root element standard attributes found in all the OTA payload documents, and the response options of returning the indication of Success, Warning or Errors in processing the request. The response message may include Warnings from business processing rules or Errors if the request did not succeed.

OTA_CourseAvailRS - Main Response element.

OTA_PayloadStdAttributes

Success

Warnings

Warning

- *Type* - Type of warning. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing)
- *Code* - The code assigned to the warnin
- *DocURL*>
- *Status*
- *Tag*

Errors

Error

- *Type* - Type of error. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing)
- *Code* - The code assigned to the warning
- *DocURL*
- *Status*
- *Tag*

GolfCourseTeeTime - The GolfCourseTeeTime attributes include the range of dates and times for which the round (or rounds) are being requested, as well as information relating to the number of golfers, the number of holes requested, and the number of times. It also includes the maximum price permitted in the response

- *StartDate* - The first date for which the availability request is valid
- *EndDate* - The last date for which the availability request is valid. If the request is for one day only, this will be the same as the StartDate
- *StartTime* - The earliest time for which the availability request is valid
- *EndTime* - The latest start time for which the availability request is valid. The StartTime and EndTime refer to each day of the period represented by the StartDate and EndDate. Where the start and end time varies by day, separate requests must be sent
- *NumberOfGolfers* - The number of golfers covered by this request.
- *NumberOfHoles* - The number of holes that the golfers wish to play within one round. If this differs by day, separate messages must be sent.
- *NumberOfTimes* - The number of Tee Times covered by this request. For example, if a foursome wants to play two separate rounds within one day, the number of tee times would be 2

- *MaxPrice* - The highest price acceptable for the requested rounds. Where a number of courses are represented by a system, only those rounds that are at or below the highest price should be returned
- *Currency* - The ISO currency code in which the price is to be quoted, and in which the max price is stated
- *ConfID* – A confirmation Identification issued by the supplier's authority.

Fee - The Fee is the actual amount to be charged for the associated round of golf. It includes the RateQualifier and a description of that rate, as well as the amount and currency

- *RateQualifier* - A code representing the basis for which the rate was quoted. If a discount or net rate was given, this indicates the reason for the discount or net rate
- *Name* - A description of the rate
- *Amount* - The amount of the rate.
- *Currency* - The ISO currency code for the currency in which the rate is quoted

3.5. **Course Tee Time Reservation Request Message (OTA_CourseResRQ)**

The root tag of OTA_CourseResRQ uses the external *include* <OTA_v2ent.xsd> that defines the root element standard attributes found in all the OTA payload documents

OTA_CourseResRQ - The OTA_CourseResRQ message is used to request a reservation at a known single course for one or more potential tee times. The specific information about the golfer or golfers is necessary in order to validate booking rules and set rates. Where the booking entity has the authority to take a reservation without a request (from an existing block) then the Notification boolean will be set to "Yes". It is the main Request element.

OTA_PayloadStdAttributes - See above

GolfCourseRes - A specific reservation for a tee time for an individual or a number of golfers

- *GolfCourseID* - The Unique Identifier associated with the golf course
- *Notification* - Where the booking entity has the authority to take a reservation without a request (from an existing block) then the Notification boolean will be set to "Yes". In this case, the represented reservation will be considered to have been made, and this message is a notification of the reservation. The supplier has contractually agreed to accept such a reservation, and cannot deny the notification.
- *Id* - A unique identifier assigned to this reservation
- *Status* - The status of this reservation transaction
- *StatusMessage* - A description or message associated with the status
- *RequestorResId* - The unique Identifier used by the party booking the reservation
- *ResponderResConfId* - The unique identifier assigned by the supplier.
- *CancellationPolicy* - An explanation of the cancellation policy in effect for this reservation
- *CancellationWindow* - The final time at which a cancellation may be made
- *CancellationCharge* - The charge to be applied in the event of cancellation

Round - A single round of golf that may include one or more golfers. If there is more than one round, each round will be represented separately

- *RoundId* - A unique identifier applied to this round
- *PlayDate* - The date for which the round is reserved
- *PlayTime* - The time for which the round is reserved
- *NumberOfGolfers* - The number of golfers represented in this round
- *Packageld* - An identification of a package of which this round is a part
- *StartingTee* - The tee on which the round will start. This could be a hole number, or a course designation where there are multiple courses.
- *NumberOfCarts* - The number of carts included within the reservation
- *TotalAmount* - The Total Amount for this round
- *Currency* - The ISO currency code in which the TotalAmount is represented.

Charge - The actual pricing and costing information about the charges being applied to the round. One or more Charge may apply to a single round

- *Id* - A unique identifier representing the charge
- *Description* - A description of the charge
- *Quantity* - The quantity associated with this charge
- *UnitPrice* - The unit price for this charge

- *UnitCost* - A unit cost associated with this charge. The cost to the booking party may be different from the charge to the golfer
- *Amount* - Total amount extended
- *BillingType* - A code representing the method of billing between parties

Golfer - Information about the individual golfer for which the reservation is being requested

- *Gender* – The Gender (if known) of the golfer.
- *DateOfBirth* – The Date of Birth (if known) of the golfer.

Memberships – a collection of membership information for the golfer.

Membership – information about a specific membership.

- *Code* – The code representing the membership.
- *ID* – The identification of this member within the membership group.

PaymentForm – See OTA_Profile for description.

PersonName – See OTA_Profile for description

Address – See OTA_Profile for description

Telephone – See OTA_Profile for description

Email – See OTA_Profile for description

RateQualifier – A single or multiple codes. A RateQualifier notes a criterion that was used for setting the actual charge as applied to the round. For example, if the requestor is a wholesaler or reseller, the RateQualifier will be a code associated with that status. If the individual golfer is a member of an organization that typically gets a discount (such as NGCOA, PGA, USGA, etc.) then that code would be included in the rate qualifier. Twilight, weekday, local, hotel guest, etc. will also be reflected here.

3.6. **Course Tee Time Reservation Response Message (OTA_CourseResRS)**

The root tag of OTA_CourseResRS uses the external *include* OTA_v2ent.xsd that defines the root element standard attributes found in all the OTA payload documents, and the response options of returning the indication of Success, Warning or Errors in processing the request. The response message may include Warnings from business processing rules or Errors if the request did not succeed.

OTA_CourseResRS - The OTA_CourseResRS message is the response to a reservation request at a known single course for one or more tee times. The specific information about the golfer or golfers is optional where that data has been supplied in the request. Where the booking entity has the authority to take a reservation without a request (from an existing block) and the Notification boolean has been set to "Yes", only the attributes of GolfCourseRes are required. All elements are optional. It is the main Response element.

OTA_PayloadStdAttributes - See above

Success

Warnings

Warning

- *Type* - Type of warning. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing)
- *Code* - The code assigned to the warnin
- *DocURL*>
- *Status*
- *Tag*

Errors

Error

- *Type* - Type of error. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing)
- *Code* - The code assigned to the warning
- *DocURL*
- *Status*
- *Tag*

GolfCourseRes - A specific reservation for a tee time for an individual or a number of golfers. See OTA_CoursResRQ for description.

3.7. **Sample XML Message for Course Search Request**

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_CourseSearchRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opentravel.org/OTA OTA_CourseSearchRQ.xsd" EchoToken="12345"
TimeStamp="1999-05-31T13:20:00-05:00" Target="Production" Version="2001C" SequenceNmbr="123456"
GolfCourseID="" DetailResponse="true">
  <Criteria Name="Architect" Value="Robert Trent Jones" Required="true"/>
  <Criteria Name="Singles Confirmed" Value="Yes" Required="true"/>
  <Criteria Name="ADA Challenged" Value="Wheelchair" Required="true"/>
  <Criteria Name="Slope" Value="110" Operation="LessThan" Required="true"/>
</OTA_CourseSearchRQ>
```

3.8. Sample XML Message for Course Search Response

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_CourseSearchRS xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opentravel.org/OTA OTA_CourseSearchRS.xsd" EchoToken="12345"
TimeStamp="1999-05-31T13:20:00-05:00" Target="Production" Version="2001C" SequenceNmbr="123456"
DetailResponse="true">
  <Success/>
  <GolfCourse Id="FL1234" Name="Sea Grass Golf Resort" City="Jupiter" State="FL" Country="USA"
PostalCode="21921" County="Palm Beach" Province="" PhoneNumber="444 444-4444">
    <Trait Name="Architect" Value="Robert Trent Jones"/>
    <Trait Name="Singles Confirmed" Value="Yes"/>
    <Trait Name="ADA Challenged" Value="Wheelchair"/>
    <Trait Name="Slope" Value="110"/>
    <Trait Name="Metal Spikes" Value="No"/>
    <Trait Name="Caddies Available" Value="No"/>
    <Trait Name="Yardage" Value="6345"/>
    <Trait Name="Personal Carts Permitted" Value="No"/>
    <Trait Name="Fivesome" Value="No"/>
    <Trait Name="GrassType" Value="Bermuda"/>
  </GolfCourse>
  <GolfCourse Id="FL4321" Name="Beach Side Golf Resort" City="Palm Beach Gardens" State="FL" Country="USA"
PostalCode="21932" County="Palm Beach" Province="" PhoneNumber="555 555-5555">
    <Trait Name="Architect" Value="Jack Nicklaus"/>
    <Trait Name="Singles Confirmed" Value="Yes"/>
    <Trait Name="ADA Challenged" Value="Blind"/>
    <Trait Name="Slope" Value="112"/>
    <Trait Name="Metal Spikes" Value="Yes"/>
    <Trait Name="Caddies Available" Value="Yes"/>
    <Trait Name="Yardage" Value="7102"/>
    <Trait Name="Fivesome" Value="Yes"/>
    <Trait Name="GrassType" Value="Rye"/>
  </GolfCourse>
</OTA_CourseSearchRS>
```

3.8.1. Sample XML Message for Course Availability Request

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_CourseAvailRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opentravel.org/OTA OTA_CourseAvailRQ.xsd" EchoToken="12345"
TimeStamp="1999-05-31T13:20:00-05:00" Target="Production" Version="2001C" SequenceNmbr="123456"
GolfCourseID="FL1234">
  <GolfCourseTeeTime StartDate="2001-10-31" EndDate="2001-10-31" StartTime="13:00:00" EndTime="14:30:00"
NumberOfGolfers="4" NumberOfHoles="18" NumberOfTimes="1" MaxPrice="80" Currency="USD">
    <RateQualifier>NGCOA</RateQualifier>
    <Memberships>
      <Membership/>
    </Memberships>
  </GolfCourseTeeTime>
</OTA_CourseAvailRQ>
```

3.9. Sample XML Message for Course Availability Response

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_CourseAvailRS xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opentravel.org/OTA OTA_CourseAvailRS.xsd" EchoToken="12345"
TimeStamp="1999-05-31T13:20:00-05:00" Target="Production" Version="2001C" SequenceNmbr="123456"
GolfCourseID="FL1234">
  <Success/>
  <GolfCourseTeeTime StartDate="2001-10-31" EndDate="2001-10-31" StartTime="13:00:00" EndTime="14:30:00"
NumberOfGolfers="4" NumberOfHoles="18" NumberOfTimes="1" MaxPrice="80" Currency="USD">
    <Fee RateQualifier="NGCOA" Name="GreensFee" Amount="70" Currency="USD"/>
    <Fee RateQualifier="NGCOA" Name="CartFee" Amount="11" Currency="USD"/>
  </GolfCourseTeeTime>
</OTA_CourseAvailRS>
```

3.10. Sample XML Message for Course Reservation Request

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_CourseResRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA OTA_CourseResRQ.xsd" EchoToken="12345" TimeStamp="1999-
05-31T13:20:00-05:00" Target="Production" Version="2001C" SequenceNmbr="123456" GolfCourseID="FL1234"
Notification="true">
  <GolfCourseRes Id="123456" Status="New" StatusMessage="New Reservation Request" RequestorResId="AC4132"
ResponderResConfId="11223344" CancellationPolicy="Standard" CancellationWindow="2001-09-11T09:30:47-05:00"
CancellationCharge="55.00">
    <Round RoundId="23456" PlayDate="2002-02-10" PlayTime="11:06:00" NumberOfGolfers="2" PackageId=""
StartingTee="1" NumberOfCarts="1" TotalAmount="140.00" Currency="USD">
      <Charge Id="Std" Description="Standard Rate - Tourist" Quantity="2" UnitPrice="70" UnitCost="63.34"
Amount="140.00" BillingType="Net"/>
      <Golfer Gender="M" DateOfBirth="1923-06-03">
        <Memberships>
          <Membership Type="NGCOA" ID="4123546"/>
        </Memberships>
        <PaymentForm CostCenterId="String" RPH="String">
          <CreditCard CardType="Credit" CreditCardCode="AX" CardNumber="31452343568762231"
SeriesCode="1122334" EffectiveDate="1967-08-13" ExpireDate="2004-08-13">
            <CardHolderName>
              <NamePrefix>Mr.</NamePrefix>
              <GivenName>Bobby</GivenName>
              <Surname>Jones</Surname>
              <PrivacyDetails ShareSynchInd="Yes" ShareMarketInd="Yes"/>
            </CardHolderName>
            <CardIssuerName IssuingBankId="1122334455"/>
            <Address AddressType="Home" DefaultInd="Yes">
              <StreetNmbr PO_Box="String">123 Augusta Lane</StreetNmbr>
              <BldgRoom>String</BldgRoom>
              <AddressLine ParsedInd="1"/>
              <CityName PostalCode="23456">Atlanta</CityName>
              <StateProv StateCode="GA">Georgia</StateProv>
            </Address>
          </CreditCard>
        </PaymentForm>
      </Golfer>
    </Round>
  </GolfCourseRes>
</OTA_CourseResRQ>
```

```
        <CountryName CountryCode="US">United States</CountryName>
        <PrivacyDetails ShareSynchInd="Yes" ShareMarketInd="Yes"/>
    </Address>
    <PrivacyDetails ShareSynchInd="Yes" ShareMarketInd="Yes"/>
</CreditCard>
    <PrivacyDetails ShareSynchInd="Yes" ShareMarketInd="Yes"/>
</PaymentForm>
    <PersonName DefaultIndicator="1" NameType="String">
        <NamePrefix>Mr.</NamePrefix>
        <GivenName>Bobby</GivenName>
        <Surname>Jones</Surname>
        <PrivacyDetails ShareSynchInd="Yes" ShareMarketInd="Yes"/>
    </PersonName>
    <Address AddressType="Home" DefaultInd="Yes">
        <StreetNmbr PO_Box="String">123 Augusta Lane</StreetNmbr>
        <BldgRoom>String</BldgRoom>
        <AddressLine ParsedInd="1"/>
        <CityName PostalCode="23456">Atlanta</CityName>
        <StateProv StateCode="GA">Georgia</StateProv>
        <CountryName CountryCode="US">United States</CountryName>
        <PrivacyDetails ShareSynchInd="Yes" ShareMarketInd="Yes"/>
    </Address>
    <Telephone PhoneTechType="String" CountryAccessCode="1" AreaCityCode="703"
PhoneNumber="444-5555">
        <PrivacyDetails ShareSynchInd="Yes" ShareMarketInd="Yes"/>
    </Telephone>
    <Email DefaultInd="Yes" EmailType="String">
        <Email>"golfer@myswing.com"</Email>
        <PrivacyDetails ShareSynchInd="Yes" ShareMarketInd="Yes"/>
    </Email>
</Golfer>
    <RateQualifier>NGCOA</RateQualifier>
</Round>
</GolfCourseRes>
</OTA_CourseResRQ>
```

3.11. Sample XML Message for Course Reservation Response

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_CourseResRS xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA OTA_CourseResRS.xsd" EchoToken="12345" TimeStamp="1999-
05-31T13:20:00-05:00" Target="Production" Version="2001C" SequenceNmbr="123456" GolfCourseID="FL1234">
  <Success/>
  <GolfCourseRes Id="123456" Status="New" StatusMessage="New Reservation Request"
RequestorResId="11223344" ResponderResConfId="11223344" CancellationPolicy="Standard"
CancellationWindow="2001-09-11T09:30:47-05:00" CancellationCharge="55.00">
    <Round RoundId="23456" PlayDate="2002-02-10" PlayTime="11:06:00" NumberOfGolfers="2" PackageId=""
StartingTee="1" NumberOfCarts="1" TotalAmount="140.00" Currency="USD">
      <Charge Id="Std" Description="Standard Rate - Tourist" Quantity="2" UnitPrice="70" UnitCost="63.34"
Amount="140.00" BillingType="Net"/>
      <Golfer Gender="M" DateOfBirth="1923-06-03">
        <Memberships>
          <Membership Type="NGCOA" ID="4123546"/>
        </Memberships>
        <PaymentForm CostCenterId="String" RPH="String">
          <CreditCard CardType="Credit" CreditCardCode="AX" CardNumber="31452343568762231"
SeriesCode="1122334" EffectiveDate="1967-08-13" ExpireDate="2004-08-13">
            <CardHolderName>
              <NamePrefix>Mr.</NamePrefix>
              <GivenName>Bobby</GivenName>
              <Surname>Jones</Surname>
              <PrivacyDetails ShareSynchInd="Yes" ShareMarketInd="Yes"/>
            </CardHolderName>
            <CardIssuerName IssuingBankId="1122334455"/>
            <Address AddressType="Home" DefaultInd="Yes">
              <StreetNmbr PO_Box="String">123 Augusta Lane</StreetNmbr>
              <BldgRoom>String</BldgRoom>
              <AddressLine ParsedInd="1"/>
              <CityName PostalCode="23456">Atlanta</CityName>
              <StateProv StateCode="GA">Georgia</StateProv>
              <CountryName CountryCode="US">United States</CountryName>
              <PrivacyDetails ShareSynchInd="Yes" ShareMarketInd="Yes"/>
            </Address>
            <PrivacyDetails ShareSynchInd="Yes" ShareMarketInd="Yes"/>
          </CreditCard>
          <PrivacyDetails ShareSynchInd="Yes" ShareMarketInd="Yes"/>
        </PaymentForm>
        <PersonName DefaultIndicator="1" NameType="String">
          <NamePrefix>Mr.</NamePrefix>
          <GivenName>Bobby</GivenName>
          <Surname>Jones</Surname>
          <PrivacyDetails ShareSynchInd="Yes" ShareMarketInd="Yes"/>
        </PersonName>
        <Address AddressType="Home" DefaultInd="Yes">
          <StreetNmbr PO_Box="String">123 Augusta Lane</StreetNmbr>
          <BldgRoom>String</BldgRoom>
          <AddressLine ParsedInd="1"/>
          <CityName PostalCode="23456">Atlanta</CityName>
          <StateProv StateCode="GA">Georgia</StateProv>
          <CountryName CountryCode="US">United States</CountryName>
          <PrivacyDetails ShareSynchInd="Yes" ShareMarketInd="Yes"/>
        </Address>
        <Telephone PhoneTechType="String" CountryAccessCode="1" AreaCityCode="703"
PhoneNumber="444-5555">
          <PrivacyDetails ShareSynchInd="Yes" ShareMarketInd="Yes"/>
        </Telephone>
        <Email DefaultInd="Yes" EmailType="String">
          <Email>"golfer@myswing.com"</Email>
          <PrivacyDetails ShareSynchInd="Yes" ShareMarketInd="Yes"/>
        </Email>
      </Golfer>
    </RateQualifier>NGCOA</RateQualifier>
  </GolfCourseRes>
</OTA_CourseResRS>
```

```
</Round>  
</GolfCourseRes>  
</OTA_CourseResRS>
```

3.12. *Revision History*

<u>Date</u>	<u>Version</u>	<u>Revisions</u>
November 5, 2001	0.1	Initial Draft.