



HOTEL TECHNOLOGY – NEXT GENERATION  
Property/Distribution Solution  
Usage Profile – Availability  
Version 1.0.4

FINAL

Version 1.0.4 includes a minor update to the 'About HTNG' section.

Copyright 2007, Hotel Technology Next Generation

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the copyright owner.

For any software code contained within this specification, permission is hereby granted, free-of-charge, to any person obtaining a copy of this specification (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the above copyright notice and this permission notice being included in all copies or substantial portions of the Software.

Following paragraph added, for v1.04 on 13 May 2010:

Manufacturers and software providers shall not claim compliance with portions of the requirements of any HTNG specification or standard, and shall not use the HTNG name or the name of the specification or standard in any statements about their respective product(s) unless the product(s) is (are) certified as compliant to the specification or standard.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES, OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF, OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Permission is granted for implementers to use the names, labels, etc. contained within the specification. The intent of publication of the specification is to encourage implementations of the specification.

This specification has not been verified for avoidance of possible third-party proprietary rights. In implementing this specification, usual procedures to ensure the respect of possible third-party intellectual property rights should be followed.

The names Hotel Technology Next Generation and HTNG, and logos depicting these names, are trademarks of Hotel Technology Next Generation. Permission is granted for implementers to use the aforementioned names in technical documentation for the purpose of acknowledging the copyright and including the notice required above. All other use of the aforementioned names and logos requires the permission of Hotel Technology Next Generation, either in written form or as explicitly permitted for the organizations members through the current terms and conditions of membership.

## TABLE OF CONTENTS

1.0	INTRODUCTION.....	1
1.1	Purpose.....	1
1.2	Scope.....	1
1.3	Overview.....	1
1.4	References.....	1
1.5	Terminology.....	1
1.6	Assumptions and Dependencies .....	2
2.0	USAGE PROFILE 1: UPDATE AVAILABILITY .....	3
2.1	Scope.....	3
2.2	Update Room Type Availability Request .....	3
2.2.1	<i>Usage Profile Table</i> .....	3
2.2.2	<i>Example Message</i> .....	5
2.3	Update Rate Availability Request.....	7
2.3.1	<i>Usage Profile Table</i> .....	7
2.3.2	<i>Example Message</i> .....	9
2.4	Update Room/Rate Availability Request.....	10
2.4.1	<i>Usage Profile Table</i> .....	10
2.4.2	<i>Example Message</i> .....	13
2.5	Update Segment Availability Request .....	14
2.5.1	<i>Usage Profile Table</i> .....	14
2.5.2	<i>Example Message</i> .....	16
2.6	Update Segment/Room Availability Request .....	17
2.6.1	<i>Usage Profile Table</i> .....	18
2.6.2	<i>Example Message</i> .....	20
2.7	Update House Availability Request.....	22
2.7.1	<i>Usage Profile Table</i> .....	22
2.7.2	<i>Example Message</i> .....	24
2.8	Update Hurdle Rates Request .....	25
2.8.1	<i>Usage Profile Table</i> .....	26
2.8.2	<i>Example Message</i> .....	28
2.9	Update Availability Respose.....	29
2.9.1	<i>Usage Profile Table</i> .....	29

2.9.2	<i>Example Message</i> .....	30
2.10	FAQ.....	31
2.10.1	<i>How should errors and warnings be used?</i> .....	31

## PREFACE

Hotel Technology Next Generation (“HTNG”) is a nonprofit organization with global scope, formed in 2002 to facilitate the development of next-generation, customer-centric technologies to better meet the needs of the global hotel community. HTNG’s mission is to provide leadership that will facilitate the creation of one (or more) industry solution set(s) for the lodging industry that:

- Are modeled around the customer and allow for a rich definition and distribution of hotel products, beyond simply sleeping rooms;
- Comprise best-of-breed software components from existing vendors, and enable vendors to collaboratively produce world-class software products encompassing all major areas of technology spending: hotel operations, telecommunications, in-room entertainment, customer information systems, and electronic distribution;
- Properly exploit and leverage a base system architecture that provides integration and interoperability through messaging; and that provides security, redundancy, and high availability;
- Target the needs of hotel companies up to several hundred properties, that are too small to solve the issues themselves;
- Will reduce technology management cost and complexity while improving reliability and scalability; and
- Can be deployed globally, managed remotely, and outsourced to service providers where needed.

In June 2005, HTNG announced the first-ever “Branding and Certification Program” for hotel technology. This program will enable vendors to certify their products against open HTNG specifications, and to use the "HTNG Certified" logo in their advertising and collateral materials. It will enable hotels to determine which vendors have completed certification of their products against which specific capabilities, and the environments in which performance is certified. HTNG’s vision is to achieve a flexible technical environment that will allow multiple vendors’ systems to interoperate and that will facilitate vendor alliances and the consolidation of applications, in order to provide hotels with easily managed, continually evolving, cost-effective solutions to meet their complete technology needs on a global basis.

## ACKNOWLEDGEMENTS

HTNG gratefully acknowledges the contributions of the following people in the development of this document:

Juan Gasparini	IDEaS
Francesca Pasetti	Cendant
Jay Rosamilia	PAR SMS
Stephen Burke	HBSI
Andrew Rutter	Nirvana
Ken Kuhn	Marriott International
Mark Houser	MSI
Keith Cook	Pegasus
Pieter Hugo	Pegasus
Bernd Jändl	Amadeus Hospitality

## 1.0 INTRODUCTION

### 1.1 Purpose

This document defines the Hotel Technology Next Generation (“HTNG”) Usage Profiles for Reservation Messages for Property Distribution based on the OpenTravel Alliance (“OTA”) specifications published by this organization.

### 1.2 Scope

This document defines a common HTNG implementation of the OTA specifications for reservation messages and is designed as a guide for project managers, programmers, and analysts to gain detailed information needed to implement these messages.

### 1.3 Overview

The messages in the OTA specification cover a large number of optional fields and the same information could be transferred in a number of different fields.

The intent of the HTNG usage profiles is to recommend a minimum common denominator and clarify what fields should be used to transfer the data required. The main aim is to avoid having to pass the same information in more than one field, thus avoiding confusion.

Trading partners may agree prior to implementation to use additional fields for data not covered in the usage profile, including TPA extensions as per the OTA specifications schema.

A note on Min/Max LOS: the method listed in the recommendations is the most basic method of passing Min/Max length of stay and therefore it should be the recommended way. Full LOS patterns can be sent via the OTA message and upon agreement between the implementing partners this method could be the one used instead of the recommended one.

### 1.4 References

The following documents are referenced in this document:

1. The OTA Specifications, published twice yearly and available at <http://www.opentravel.org/>

### 1.5 Terminology

Table 1 defines terms and acronyms used throughout this document.

**Table 1 - Terminology**

Term	Definition
HTNG	Hotel Technology – Next Generation
OTA	OpenTravel Alliance

Term	Definition
XML	Extensible Markup Language – This is a general-purpose markup language for creating special- purpose markup languages, capable of describing different kinds of data.

#### 1.6 Assumptions and Dependencies

It is assumed the usage profiles provide a common starting point for the definition of the messages exchanged and that the implementers of these messages will:

- a) Add expansions as needed to enable exchange of additional information while retaining compliance with the OTA specifications.
- b) Report expansions deemed common to HTNG for consideration as additional usage profiles.
- c) Report any missing elements or attributes to OTA for inclusion in a future specification release.



## 2.0 USAGE PROFILE 1: UPDATE AVAILABILITY

The update availability usage profile covers the pushing of availability settings from one system that defines them or provides a user an interface to define them or receives them from another system to another system with the ability to book or change a reservation.

### 2.1 Scope

This usage profile includes any availability settings that can be pushed from one system to another.

It does not include polling availability settings.

### 2.2 Update Room Type Availability Request

The update room type availability request is sent using the OTA\_HotelAvailNotifRQ message.

#### 2.2.1 Usage Profile Table

Element   @Attribute	Num	Description/Contents
OTA_HotelAvailNotifRQ	1	Root element of the message.
@EchoToken	0..1	Optional
@TimeStamp	1	time of the transaction
@Version	1	Version is a mandatory attribute in OTA – therefore it must remain mandatory in HTNG in order to be able to use the same message
@TransactionIdentifier	0..1	Optional. If the availability update is the result of a reservation upload, the reservation number should be indicated in this field.
@MessageContentCode	1	The attribute refers to OTA code list MCC which includes RateAvail, RoomAvail, RoomRateAvail, SegmentAvail, SegmentRoomAvail, HouseAvail, HurdleRateUpdate. For room type availability this should be set to 1.
OTA_HotelAvailNotifRQ / AvailStatusMessages	1	Must be sent for the message to have a meaning
@HotelCode	1	This is the code of the property whose availability is being updated

Element   @Attribute	Num	Description/Contents
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / UniqueID	1	
@Type	1	Refers to OTA code list UIT – nr 16 = Reference. This is used to identify each single availability status message for error reporting purposes..
@ID	1	A unique incremental number for each availability message that identifies that specific message.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / StatusApplicationControl	1	Element defining application of controls being sent.
@Start, @End	1	The first and last dates for which the availability update is being sent
@Mon,@Tue,@Weds,@Thur,@Fri, @Sat,@Sun	0..1	The day of the week indicators are used to communicate which days of the week the update pertains to. If one is sent they must all be sent.
@InvTypeCode	1	This is the room type code for which the update is being sent.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage	0..n	Optional. Although all restrictions are optional, at least one should be sent for the message to have any meaning. Upper limit to be defined by trading partners.
@BookingLimit	0..1	This is the maximum number of rooms for the InvTypeCode sent that can be booked. Although all restrictions are optional, at least one should be sent for the message to have any meaning.
@BookingLimitMessageType	0..1	Enumerated values are used to indicate whether the booking limit sent in the transmission is used to set, adjust or delete the booking limit. (SetLimit, AdjustLimit, RemoveLimit)
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / RestrictionStatus	0..1	Optional. If no BookingLimit or Length of Stay restriction is sent, then the Status field must be sent for the message to be meaningful.
@Status	0..1	Optional. Enumeration; possible values ="Open", "Close", "ClosedOnArrival", "OnRequest". If one of the partners implementing the message does not support all the enumerations, the enumerations that are not supported should be mapped to supported enumerations by the receiving system. An un-supported enumeration would be ignored by the receiving system – therefore the type of status messages must be agreed upon implementation between the 2 partners.

Element   @Attribute	Num	Description/Contents
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay	0..1	Optional. If no BookingLimit or Status restriction is sent, then the LengthsOfStay field must be sent with some LOS restrictions in order for the message to be meaningful.
@FixedPatternLength	0..1	Optional. If both partners agree to send the Full Pattern LOS then this field is set to the length of the pattern string further inside the message. For example, this would be set to 7 when the string has seven characters, from LOS1 to LOS7+.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay / LengthOfStay	0..n	Conditionally mandatory. If LengthsOfStay is sent then there must be at least one LengthOfStay element.
@MinMaxMessageType	0..1	MinMaxLengthOfStay set = SetMinLOS when sending MinLOS, the default setting. It can be set = FullPatternLOS if both partners support the Full Pattern Length of Stay as available in the OTA message. Other values could be used based on partner agreement.
@Time, @TimeUnit	0..1	If sending MinLOS then TimeUnit set = Day and Time set to the MinLOS value.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay / LengthOfStay / LOS_Pattern	0..1	
@FullPatternLOS	0..1	This is set to the string with 'Y' for the LOS that are open and 'N' for the LOS that are closed. The length of the string is indicated in @FixedPatternLength described above. For example, to open the odd LOS (LOS1,LOS3,LOS5 and LOS7) and close the even LOS (LOS2, LOS4 and LOS6) then the string would be "YNYNYNY"

### 2.2.2 Example Message

## Scenario:

PMS “ABC” needs to transmit the following ROOM TYPE availability update to CRS “123”

Hotel Code	HXCAIZZ
Start date:	Jan 1 <sup>st</sup> 2006
End date:	Jan 14 <sup>th</sup> 2006
Change applies to DOW:	FRI-SAT-SUN
Room Type:	A1K
Number allotted:	10
Status:	On Request
Min LOS	2

```
<OTA_HotelAvailNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05 "
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance "
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA_HotelAvailNotifRQ.xsd" Version="1.001 "
EchoToken="54638383" TransactionIdentifier="938383" TimeStamp="2005-08-01T09:30:47-05:00 "
MessageContentCode="1">
  <!-- MessageContentCode=1 for Room Type Availability-->
  <AvailStatusMessages HotelCode="HXCAIZZ">
    <AvailStatusMessage BookingLimit="10" BookingLimitMessageType="SetLimit">
      <StatusApplicationControl Start="2006-01-01" End="2006-01-14" InvTypeCode="A1K" Mon="0 "
Tue="0" Weds="0" Thur="0" Fri="1" Sat="1" Sun="1">
        </StatusApplicationControl>
      <LengthsOfStay>
        <LengthOfStay MinMaxMessageType="SetMinLOS" Time="2" TimeUnit="Day">
          </LengthOfStay>
        </LengthsOfStay>
      <UniqueID Type="16" ID="1"></UniqueID>
      <RestrictionStatus Status="OnRequest"></RestrictionStatus>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailNotifRQ>
```

### 2.3 Update Rate Availability Request

The update rate availability request is sent using the OTA\_HotelAvailNotifRQ message.

#### 2.3.1 Usage Profile Table

Element   @Attribute	Num	Description/Contents
OTA_HotelAvailNotifRQ	1	Root element of the message.
@EchoToken	0..1	Optional
@TimeStamp	1	time of the transaction
@Version	1	Version is a mandatory attribute in OTA – therefore it must remain mandatory in HTNG in order to be able to use the same message
@TransactionIdentifier	0..1	Optional. If the availability update is the result of a reservation upload, the reservation number should be indicated in this field.
@MessageContentCode	1	The attribute refers to OTA code list MCC. For rate availability this should be set to 2.
OTA_HotelAvailNotifRQ / AvailStatusMessages	1	Must be sent for the message to have a meaning
@HotelCode	1	This is the code of the property whose availability is being updated
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / UniqueID	1	
@Type	1	Refers to OTA code list UIT – nr 16 = Reference. This is used to identify each single availability status message for error reporting purposes..
@ID	1	A unique incremental number for each availability message that identifies that specific message.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / StatusApplicationControl	1	Element defining application of controls being sent.

Element   @Attribute	Num	Description/Contents
@Start, @End	1	The first and last dates for which the availability update is being sent
@Mon, @Tue, @Weds, @Thur, @Fri, @Sat, @Sun	0..1	The day of the week indicators are used to communicate which days of the week the update pertains to. If one is sent they must all be sent.
@RatePlanCategory	0..1	This is the Category or segment which the rate whose availability is being updated belongs to.
@RatePlanCode	1	This is the rate plan whose availability is being updated.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage	0..n	Optional. Although all restrictions are optional, at least one should be sent for the message to have any meaning. Upper limit to be defined by trading partners.
@BookingLimit	0..1	This is the maximum number of rooms for the RatePlanCode sent that can be booked.
@BookingLimitMessageType	0..1	Enumerated values are used to indicate whether the booking limit sent in the transmission is used to set, adjust or delete the booking limit. (SetLimit, AdjustLimit, RemoveLimit)
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / RestrictionStatus	0..1	Optional. If no BookingLimit or Length of Stay restriction is sent, then the Status field must be sent for the message to be meaningful.
@Status	0..1	Enumeration; possible values ="Open", "Close", "ClosedOnArrival", "OnRequest". If one of the partners implementing the message does not support all the enumerations, the enumerations that are not supported should be mapped to supported enumerations by the receiving system. An un-supported enumeration would be ignored by the receiving system – therefore the type of status messages must be agreed upon implementation between the 2 partners.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay	0..1	Optional. If no BookingLimit or Status restriction is sent, then the LengthsOfStay field must be sent with some LOS restrictions in order for the message to be meaningful.
@FixedPatternLength	0..1	Optional. If both partners agree to send the Full Pattern LOS then this field is set to the length of the pattern string further inside the message. For example, this would be set to 7 when the string has seven characters, from LOS1 to LOS7+.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage /	0..n	Conditionally mandatory. If LengthsOfStay is sent then there must be at least one LengthOfStay element.

Element   @Attribute	Num	Description/Contents
LengthsOfStay / LengthOfStay		
@MinMaxMessageType	0..1	MinMaxLengthOfStay set = SetMinLOS when sending MinLOS, the default setting. It can be set = FullPatternLOS if both partners support the Full Pattern Length of Stay as available in the OTA message. Other values could be used based on partner agreement.
@Time, @TimeUnit	0..1	Optional. If sending MinLOS then TimeUnit set = Day and Time set to the MinLOS value.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay / LengthOfStay / LOS_Pattern	0..1	
@FullPatternLOS	0..1	If sending Full Pattern LOS then this is set to the string with 'Y' for the LOS that are open and 'N' for the LOS that are closed. The length of the string is indicated in @FixedPatternLength described above. For example, to open the odd LOS (LOS1,LOS3,LOS5 and LOS7) and close the even LOS (LOS2, LOS4 and LOS6) then the string would be "YNYNYNY"

### 2.3.2 Example Message

Scenario:

PMS "ABC" needs to transmit the following RATE availability update to CRS "123"

Hotel Code	HXCAIZZ
Start date:	Jan 1 <sup>st</sup> 2006
End date:	Jan 10 <sup>th</sup> 2006
Change applies to DOW:	SAT-SUN-MON
Rate segment:	C
Rate Plan:	CR
Number allotted:	15
Status:	Open

```

<OTA_HotelAvailNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA_HotelAvailNotifRQ.xsd" Version="1.001"
EchoToken="54638383" TransactionIdentifier="938384" TimeStamp="2005-08-01T09:30:47-05:00"
MessageContentCode="2">
<!-- MessageContentCode=2 for Rate Availability-->
  <AvailStatusMessages HotelCode="HXCAIZZ">
    <AvailStatusMessage BookingLimit="15" BookingLimitMessageType="SetLimit">
<StatusApplicationControl Start="2006-01-01" End="2006-01-10" RatePlanCategory="C" RatePlanCode="CR"
Mon="1" Tue="0" Weds="0" Thur="0" Fri="0" Sat="1" Sun="1">
    </StatusApplicationControl>
    <UniqueID Type="16" ID="1"></UniqueID>
    <RestrictionStatus Status="Open"></RestrictionStatus>
  </AvailStatusMessage>
</AvailStatusMessages>
</OTA_HotelAvailNotifRQ>

```

## 2.4 Update Room/Rate Availability Request

The update room/rate availability request is sent using the OTA\_HotelAvailNotifRQ message.

### 2.4.1 Usage Profile Table

Element   @Attribute	Num	Description/Contents
OTA_HotelAvailNotifRQ	1	Root element of the message.
@EchoToken	0..1	Optional
@TimeStamp	1	time of the transaction
@Version	1	Version is a mandatory attribute in OTA – therefore it must remain mandatory in HTNG in order to be able to use the same message
@TransactionIdentifier	0..1	Optional. If the availability update is the result of a reservation upload, the reservation number should be indicated in this field.



Element   @Attribute	Num	Description/Contents
@MessageContentCode	1	The attribute refers to OTA code list MCC which includes RateAvail, RoomAvail, RoomRateAvail, SegmentAvail, SegmentRoomAvail, HouseAvail, HurdleRateUpdate. For room rate availability this should be set to 3.
OTA_HotelAvailNotifRQ / AvailStatusMessages	1	Must be sent for the message to have a meaning
@HotelCode	1	This is the code of the property whose availability is being updated
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / UniqueID	1	
@Type	1	Refers to OTA code list UIT – nr 16 = Reference. This is used to identify each single availability status message for error reporting purposes..
@ID	1	A unique incremental number for each availability message that identifies that specific message.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / StatusApplicationControl	1	Element defining application of controls being sent.
@Start, @End	1	The first and last dates for which the availability update is being sent
@Mon, @Tue, @Weds, @Thur, @Fri, @Sat, @Sun	0..1	The day of the week indicators are used to communicate which days of the week the update pertains to. If one is sent they must all be sent.
@RatePlanCode	1	This is the rate plan whose availability is being updated.
@InvTypeCode	1	This is the Room Type which makes up the room/rate combination for which availability is being updated.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage	0..n	Optional. Although all restrictions are optional, at least one should be sent for the message to have any meaning. Upper limit to be defined by trading partners.
@BookingLimit	0..1	This is the maximum number of rooms for the room/rate combination that can be booked. If no Status or LengthOfStay restriction is sent, then the BookingLimit field must be sent for the message to have any meaning. So, for each Room/rate combination, at least one restriction between BookingLimit, Status and LengthOfStay must be sent.
@BookingLimitMessageType	0..1	Enumerated values are used to indicate whether the booking limit sent in the transmission is used to set, adjust

Element   @Attribute	Num	Description/Contents
		or delete the booking limit. (SetLimit, AdjustLimit, RemoveLimit)
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / RestrictionStatus	0..1	Optional. If no BookingLimit or LengthOfStay restriction is sent, then the Status field must be sent for the message to be meaningful.
@Status	0..1	Enumeration; possible values = "Open", "Close", "ClosedOnArrival", "OnRequest". If one of the partners implementing the message does not support all the enumerations, the enumerations that are not supported should be mapped to supported enumerations by the receiving system. An un-supported enumeration would be ignored by the receiving system – therefore the type of status messages must be agreed upon implementation between the 2 partners.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay	0..1	Optional. If no BookingLimit or Status restriction is sent, then the LengthsOfStay field must be sent with some LOS restrictions in order for the message to be meaningful.
@FixedPatternLength	0..1	Optional. If both partners agree to send the Full Pattern LOS then this field is set to the length of the pattern string further inside the message. For example, this would be set to 7 when the string has seven characters, from LOS1 to LOS7+.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay / LengthOfStay	0..n	Conditionally mandatory. If LengthsOfStay is sent then there must be at least one LengthOfStay element.
@MinMaxMessageType	0..1	MinMaxLengthOfStay set = SetMinLOS when sending MinLOS, the default setting. It can be set = FullPatternLOS if both partners support the Full Pattern Length of Stay as available in the OTA message. Other values could be used based on partner agreement.
@Time, @TimeUnit	0..1	Optional. If sending MinLOS then TimeUnit set = Day and Time set to the MinLOS value.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay / LengthOfStay / LOS_Pattern	0..1	
@FullPatternLOS	0..1	If sending Full Pattern LOS then this is set to the string with 'Y' for the LOS that are open and 'N' for the LOS that are closed. The length of the string is indicated in @FixedPatternLength described above. For example, to open the odd LOS (LOS1,LOS3,LOS5 and LOS7) and close the even LOS (LOS2, LOS4 and

Element   @Attribute	Num	Description/Contents
		LOS6) then the string would be "YNYNYNY"

#### 2.4.2 Example Message

Scenario:

PMS "ABC" needs to transmit the following ROOM/RATE availability update to CRS "123"

Hotel Code	HXCAIZZ
Start date:	Jan 1 <sup>st</sup> 2006
End date:	Jan 7 <sup>th</sup> 2006
Change applies to DOW:	MON TO FRI
Rate Plan:	CR
Room Type	A1K
Number allotted:	5
Status:	Open
Min LOS:	2

```
<OTA_HotelAvailNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05 "
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA_HotelAvailNotifRQ.xsd" Version="1.001"
EchoToken="54638383" TransactionIdentifier="938386" TimeStamp="2005-08-01T09:30:47-05:00 "
MessageContentCode="3">
  <!-- MessageContentCode=3 for Room/Rate Availability-->
  <AvailStatusMessages HotelCode="HXCAIZZ">
    <AvailStatusMessage BookingLimit="5" BookingLimitMessageType="SetLimit">
      <StatusApplicationControl Start="2006-01-01" End="2006-01-07" InvTypeCode="A1K"
RatePlanCode="CR" Mon="1" Tue="1" Weds="1" Thur="1" Fri="1" Sat="0" Sun="0">
```

```

        </StatusApplicationControl>
        <LengthsOfStay>
            <LengthOfStay MinMaxMessageType="SetMinLOS" Time="2" TimeUnit="Day">
            </LengthOfStay>
        </LengthsOfStay>
        <UniqueID Type="16" ID="1"/>
        <RestrictionStatus Status="Open"/>
    </AvailStatusMessage>
</AvailStatusMessages>
</OTA_HotelAvailNotifRQ>

```

## 2.5 Update Segment Availability Request

A Segment is also known as a Rate Group or a Rate Category. This request is sent using the OTA\_HotelAvailNotifRQ message.

### 2.5.1 Usage Profile Table

Element   @Attribute	Num	Description/Contents
OTA_HotelAvailNotifRQ	1	Root element of the message.
@EchoToken	0..1	Optional
@TimeStamp	1	time of the transaction
@Version	1	Version is a mandatory attribute in OTA – therefore it must remain mandatory in HTNG in order to be able to use the same message
@TransactionIdentifier	0..1	Optional. If the availability update is the result of a reservation upload, the reservation number should be indicated in this field.
@MessageContentCode	1	The attribute refers to OTA code list MCC which includes RateAvail, RoomAvail, RoomRateAvail, SegmentAvail, SegmentRoomAvail, HouseAvail, HurdleRateUpdate. For segment availability this should be set to 4.
OTA_HotelAvailNotifRQ / AvailStatusMessages	1	Must be sent for the message to have a meaning

Element   @Attribute	Num	Description/Contents
@HotelCode	1	This is the code of the property whose availability is being updated
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / UniqueID	1	
@Type	1	Refers to OTA code list UIT – nr 16 = Reference. This is used to identify each single availability status message for error reporting purposes..
@ID	1	A unique incremental number for each availability message that identifies that specific message.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / StatusApplicationControl	1	Element defining application of controls being sent.
@Start, @End	1	The first and last dates for which the availability update is being sent
@Mon, @Tue, @Weds, @Thur, @Fri, @Sat, @Sun	0..1	The day of the week indicators are used to communicate which days of the week the update pertains to. If one is sent they must all be sent.
@RatePlanCategory	0..1	This is the Category or segment for which availability is being updated.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage	0..n	Optional. Although all restrictions are optional, at least one should be sent for the message to have any meaning. Upper limit to be defined by trading partners.
@BookingLimit	0..1	This is the maximum number of rooms for the RatePlanCategory that can be booked. If no Status or LengthOfStay restriction is sent, then the BookingLimit field must be sent for the message to have any meaning. So, for each RatePlanCategory, at least one restriction between BookingLimit, Status and LengthOfStay must be sent.
@BookingLimitMessageType	0..1	Enumerated values are used to indicate whether the booking limit sent in the transmission is used to set, adjust or delete the booking limit. (SetLimit, AdjustLimit, RemoveLimit)
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / RestrictionStatus	0..1	Optional. If no BookingLimit or LengthOfStay restriction is sent, then the Status field must be sent for the message to be meaningful.
@Status	1	Enumeration; possible values = "Open", "Close", "ClosedOnArrival", "OnRequest". If one of the partners implementing the message does not support all the enumerations, the enumerations that are not supported

Element   @Attribute	Num	Description/Contents
		should be mapped to supported enumerations by the receiving system. An un-supported enumeration would be ignored by the receiving system – therefore the type of status messages must be agreed upon implementation between the 2 partners.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay	0..1	Optional. If no BookingLimit or Status restriction is sent, then the LengthsOfStay field must be sent with some LOS restrictions in order for the message to be meaningful.
@FixedPatternLength	0..1	Optional. If both partners agree to send the Full Pattern LOS then this field is set to the length of the pattern string further inside the message. For example, this would be set to 7 when the string has seven characters, from LOS1 to LOS7+.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay / LengthOfStay	0..n	Conditionally mandatory. If LengthsOfStay is sent then there must be at least one LengthOfStay element.
@MinMaxMessageType	0..1	MinMaxLengthOfStay set = SetMinLOS when sending MinLOS, the default setting. It can be set = FullPatternLOS if both partners support the Full Pattern Length of Stay as available in the OTA message. Other values could be used based on partner agreement.
@Time, @TimeUnit	0..1	Optional. If sending MinLOS then TimeUnit set = Day and Time set to the MinLOS value.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay / LengthOfStay / LOS_Pattern	0..1	
@FullPatternLOS	0..1	If sending Full Pattern LOS then this is set to the string with 'Y' for the LOS that are open and 'N' for the LOS that are closed. The length of the string is indicated in @FixedPatternLength described above. For example, to open the odd LOS (LOS1,LOS3,LOS5 and LOS7) and close the even LOS (LOS2, LOS4 and LOS6) then the string would be "YNYNYNY"

### 2.5.2 Example Message

Scenario:

PMS “ABC” needs to transmit the following Segment/Category availability update to CRS “123”

Hotel Code	HXCAIZZ
Start date:	Jan 1 <sup>st</sup> 2006
End date:	Jan 7 <sup>th</sup> 2006
Change applies to DOW:	SUN-MON
Rate segment:	P
Number allotted:	12
Status:	Open
Min LOS:	3

```
<OTA_HotelAvailNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA_HotelAvailNotifRQ.xsd" Version="1.001"
EchoToken="54638383" TransactionIdentifier="938385" Timestamp="2005-08-01T09:30:47-05:00"
MessageContentCode="4">
<!-- MessageContentCode=4 for Segment Availability-->
  <AvailStatusMessages HotelCode="HXCAIZZ">
    <AvailStatusMessage BookingLimit="12" BookingLimitMessageType="SetLimit">
      <StatusApplicationControl Start="2006-01-01" End="2006-01-07" RatePlanCategory="P"
Mon="1" Tue="0" Weds="0" Thur="0" Fri="0" Sat="0" Sun="1">
    </StatusApplicationControl>
    <LengthsOfStay>
      <LengthOfStay MinMaxMessageType="SetMinLOS" Time="3" TimeUnit="Day">
    </LengthOfStay>
    </LengthsOfStay>
    <UniqueID Type="16" ID="1"/>
    <RestrictionStatus Status="Open"/>
  </AvailStatusMessage>
</AvailStatusMessages>
</OTA_HotelAvailNotifRQ>
```

## 2.6 Update Segment/Room Availability Request

The update segment/room availability request is sent using the OTA\_HotelAvailNotifRQ message.

### 2.6.1 Usage Profile Table

Element   @Attribute	Num	Description/Contents
OTA_HotelAvailNotifRQ	1	Root element of the message.
@EchoToken	0..1	Optional
@TimeStamp	1	time of the transaction
@Version	1	Version is a mandatory attribute in OTA – therefore it must remain mandatory in HTNG in order to be able to use the same message
@TransactionIdentifier	0..1	Optional. If the availability update is the result of a reservation upload, the reservation number should be indicated in this field.
@MessageContentCode	1	The attribute refers to OTA code list MCC which includes RateAvail, RoomAvail, RoomRateAvail, SegmentAvail, SegmentRoomAvail, HouseAvail, HurdleRateUpdate. For segment room availability this should be set to 5.
OTA_HotelAvailNotifRQ / AvailStatusMessages	1	Must be sent for the message to have a meaning
@HotelCode	1	This is the code of the property whose availability is being updated
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / UniqueID	1	
@Type	1	Refers to OTA code list UIT – nr 16 = Reference. This is used to identify each single availability status message for error reporting purposes..
@ID	1	A unique incremental number for each availability message that identifies that specific message.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / StatusApplicationControl	1	Element defining application of controls being sent.



Element   @Attribute	Num	Description/Contents
@Start, @End	1	The first and last dates for which the availability update is being sent
@Mon, @Tue, @Weds, @Thur, @Fri, @Sat, @Sun	0..1	The day of the week indicators are used to communicate which days of the week the update pertains to. If one is sent they must all be sent.
@RatePlanCategory	0..1	This is the Segment to which the room type being updated belongs
@ InvTypeCode	1	This is the Room Type which makes up the room/rate combination for which availability is being updated.
@InvCodeApplication	1	This clarifies whether the InvTypeCode is an actual room type code OR a room grouping (for instance a room group such as "deluxe" which includes multiple room types). The only 2 enumerations allowed would be InvCode for room type and InvGroupingCode for a room group. Partners will need to agree upon implementation whether the room grouping is supported by both.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage	0..n	Optional. Although all restrictions are optional, at least one should be sent for the message to have any meaning. Upper limit to be defined by trading partners.
@BookingLimit	0..1	This is the maximum number of rooms for the room/rate combination that can be booked. If no Status or LengthOfStay restriction is sent, then the BookingLimit field must be sent for the message to have any meaning. So, for each Room/rate combination, at least one restriction between BookingLimit, Status and LengthOfStay must be sent.
@BookingLimitMessageType	0..1	Enumerated values are used to indicate whether the booking limit sent in the transmission is used to set, adjust or delete the booking limit. (SetLimit, AdjustLimit, RemoveLimit)
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / RestrictionStatus	0..1	Optional. If no BookingLimit or LengthOfStay restriction is sent, then the Status field must be sent for the message to be meaningful.
@Status	1	Enumeration; possible values = "Open", "Close", "ClosedOnArrival", "OnRequest". If one of the partners implementing the message does not support all the enumerations, the enumerations that are not supported should be mapped to supported enumerations by the receiving system. An un-supported enumeration would be ignored by the receiving system – therefore the type of status messages must be agreed upon implementation between the 2 partners.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay	0..1	Optional. If no BookingLimit or Status restriction is sent, then the LengthsOfStay field must be sent with some LOS restrictions in order for the message to be meaningful.

Element   @Attribute	Num	Description/Contents
@FixedPatternLength	0..1	Optional. If both partners agree to send the Full Pattern LOS then this field is set to the length of the pattern string further inside the message. For example, this would be set to 7 when the string has seven characters, from LOS1 to LOS7+.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay / LengthOfStay	0..n	Conditionally mandatory. If LengthsOfStay is sent then there must be at least one LengthOfStay element.
@MinMaxMessageType	0..1	MinMaxLengthOfStay set = SetMinLOS when sending MinLOS, the default setting. It can be set = FullPatternLOS if both partners support the Full Pattern Length of Stay as available in the OTA message. Other values could be used based on partner agreement.
@Time, @TimeUnit	0..1	Optional. If sending MinLOS then TimeUnit set = Day and Time set to the MinLOS value.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay / LengthOfStay / LOS_Pattern	0..1	
@FullPatternLOS	0..1	If sending Full Pattern LOS then this is set to the string with 'Y' for the LOS that are open and 'N' for the LOS that are closed. The length of the string is indicated in @FixedPatternLength described above. For example, to open the odd LOS (LOS1,LOS3,LOS5 and LOS7) and close the even LOS (LOS2, LOS4 and LOS6) then the string would be "YNYNYNY"

### 2.6.2 Example Message

Scenario:

PMS "ABC" needs to transmit the following Segment/Room availability update to CRS "123"

Hotel Code	HXCAIZZ
Rate Segment:	C
Room Type	A1K
Date / FPLOS:	Jan 1 <sup>st</sup> 2006 / YYNNYY

Date / FPLOS: Jan 2<sup>nd</sup> 2006 / YNNYYYY

Date / FPLOS: Jan 3<sup>rd</sup> 2006 / NNNYYYY

```
<OTA_HotelAvailNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA_HotelAvailNotifRQ.xsd" Version="1.001"
EchoToken="54638383" TransactionIdentifier="938387" TimeStamp="2005-08-01T09:30:47-05:00"
MessageContentCode="5">
  <!-- MessageContentCode=5 for Segment/Room Availability-->
  <AvailStatusMessages HotelCode="HXCAIZZ">
    <AvailStatusMessage>
      <StatusApplicationControl Start="2006-01-01" End="2006-01-01" InvTypeCode="A1K"
RatePlanCategory="C">
      </StatusApplicationControl>
      <LengthsOfStay FixedPatternLength="7">
        <LengthOfStay MinMaxMessageType="FullPatternLOS">
          <LOS_Pattern FullPatternLOS="YNNNYY"></LOS_Pattern>
        </LengthOfStay>
      </LengthsOfStay>
      <UniqueID Type="16" ID="1"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2006-01-02" End="2006-01-02" InvTypeCode="A1K"
RatePlanCategory="C">
      </StatusApplicationControl>
      <LengthsOfStay FixedPatternLength="7">
        <LengthOfStay MinMaxMessageType="FullPatternLOS">
          <LOS_Pattern FullPatternLOS="YNNYYYY"></LOS_Pattern>
        </LengthOfStay>
      </LengthsOfStay>
      <UniqueID Type="16" ID="2"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2006-01-03" End="2006-01-03" InvTypeCode="A1K"
RatePlanCategory="C">
      </StatusApplicationControl>
      <LengthsOfStay FixedPatternLength="7">
        <LengthOfStay MinMaxMessageType="FullPatternLOS">
          <LOS_Pattern FullPatternLOS="NNNYYYY"></LOS_Pattern>
```

```

        </LengthOfStay>
    </LengthsOfStay>
    <UniqueID Type="16" ID="3"/>
</AvailStatusMessage>
</AvailStatusMessages>
</OTA_HotelAvailNotifRQ>

```

## 2.7 Update House Availability Request

The update house availability request is sent using the OTA\_HotelAvailNotifRQ message.

### 2.7.1 Usage Profile Table

Element   @Attribute	Num	Description/Contents
OTA_HotelAvailNotifRQ	1	Root element of the message.
@EchoToken	0..1	Optional
@TimeStamp	1	time of the transaction
@Version	1	Version is a mandatory attribute in OTA – therefore it must remain mandatory in HTNG in order to be able to use the same message
@TransactionIdentifier	0..1	Optional. If the availability update is the result of a reservation upload, the reservation number should be indicated in this field.
@MessageContentCode	1	The attribute refers to OTA code list MCC which includes RateAvail, RoomAvail, RoomRateAvail, SegmentAvail, SegmentRoomAvail, HouseAvail, HurdleRateUpdate. For house availability this should be set to 6.
OTA_HotelAvailNotifRQ / AvailStatusMessages	1	Must be sent for the message to have a meaning
@HotelCode	1	This is the code of the property whose availability is being updated
OTA_HotelAvailNotifRQ / AvailStatusMessages /	0..n	Optional. Although all restrictions are optional, at least one should be sent for the message to have any meaning. Upper limit to be defined by trading partners.

Element   @Attribute	Num	Description/Contents
AvailStatusMessage		
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / UniqueID	1	
@Type	1	Refers to OTA code list UIT – nr 16 = Reference. This is used to identify each single availability status message for error reporting purposes..
@ID	1	A unique incremental number for each availability message that identifies that specific message.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / StatusApplicationControl	1	Element defining application of controls being sent.
@Start, @End	1	The first and last dates for which the availability update is being sent
@Mon, @Tue, @Weds, @Thur, @Fri, @Sat, @Sun	0..1	The day of the week indicators are used to communicate which days of the week the update pertains to. If one is sent they must all be sent.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / RestrictionStatus	0..1	Optional. If no other restriction is sent (Departure restriction, min LOS), then the Status field must be sent for the message to be meaningful.
@Status	1	Enumeration; possible values = "Open", "Close", "ClosedOnArrival", "OnRequest". If one of the partners implementing the message does not support all the enumerations, the enumerations that are not supported should be mapped to supported enumerations by the receiving system. An un-supported enumeration would be ignored by the receiving system – therefore the type of status messages must be agreed upon implementation between the 2 partners.
@Restriction	0..1	<p>Optional. If no other restriction is sent (Status, min LOS), then the Restriction field must be sent.</p> <p>In the OTA message the following enumerations are possible: Master, Arrival, Departure, NonGuarantee, TravelAgent. However the HTNG recommendation should be that only Departure and Arrival are used as they are the only one that are meaningful.</p> <p>If one of the partners implementing the message does not support one of the 2 Recommended enumerations, the un-supported enumeration would be ignored by the receiving system. For instance if the transmitting system supports both arrival and departure restriction (for this type of message) but the recipient only supports departure restrictions, then the arrival restrictions would be ignored. Partners would need to agree on this</p>

Element   @Attribute	Num	Description/Contents
		during implementation.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay	0..1	Optional. If no BookingLimit or Status restriction is sent, then the LengthsOfStay field must be sent with some LOS restrictions in order for the message to be meaningful.
@FixedPatternLength	0..1	Optional. If both partners agree to send the Full Pattern LOS then this field is set to the length of the pattern string further inside the message. For example, this would be set to 7 when the string has seven characters, from LOS1 to LOS7+.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay / LengthOfStay	0..n	Conditionally mandatory. If LengthsOfStay is sent then there must be at least one LengthOfStay element.
@MinMaxMessageType	0..1	MinMaxLengthOfStay set = SetMinLOS when sending MinLOS, the default setting. It can be set = FullPatternLOS if both partners support the Full Pattern Length of Stay as available in the OTA message. Other values could be used based on partner agreement.
@Time, @TimeUnit	0..1	Optional. If sending MinLOS then TimeUnit set = Day and Time set to the MinLOS value.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay / LengthOfStay / LOS_Pattern	0..1	
@FullPatternLOS	0..1	If sending Full Pattern LOS then this is set to the string with 'Y' for the LOS that are open and 'N' for the LOS that are closed. The length of the string is indicated in @FixedPatternLength described above. For example, to open the odd LOS (LOS1,LOS3,LOS5 and LOS7) and close the even LOS (LOS2, LOS4 and LOS6) then the string would be "YNYNYNY"

### 2.7.2 Example Message

Scenario:

PMS "ABC" needs to transmit the following HOUSE availability update to CRS "123"

Hotel Code

HXCAIZZ

Date / Authorized Capacity: Jan 1<sup>st</sup> 2006 / 350  
 Date / Authorized Capacity: Jan 2<sup>nd</sup> 2006 / 348  
 Date / Authorized Capacity: Jan 3<sup>rd</sup> 2006 / 354

```
<OTA_HotelAvailNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA_HotelAvailNotifRQ.xsd" Version="1.001"
EchoToken="54638383" TransactionIdentifier="938388" Timestamp="2005-08-01T09:30:47-05:00"
MessageContentCode="6">
  <!-- MessageContentCode=6 for House Availability-->
  <AvailStatusMessages HotelCode="HXCAIZZ">
    <AvailStatusMessage BookingLimit="350" BookingLimitMessageType="SetLimit">
      <StatusApplicationControl Start="2006-01-01" End="2006-01-01">
      </StatusApplicationControl>
      <UniqueID Type="16" ID="1"/>
    </AvailStatusMessage>
    <AvailStatusMessage BookingLimit="348" BookingLimitMessageType="SetLimit">
      <StatusApplicationControl Start="2006-01-02" End="2006-01-02">
      </StatusApplicationControl>
      <UniqueID Type="16" ID="2"/>
    </AvailStatusMessage>
    <AvailStatusMessage BookingLimit="354" BookingLimitMessageType="SetLimit">
      <StatusApplicationControl Start="2006-01-03" End="2006-01-03">
      </StatusApplicationControl>
      <UniqueID Type="16" ID="3"/>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailNotifRQ>
```

## 2.8 Update Hurdle Rates Request

The update hurdle rates request is sent using the OTA\_HotelAvailNotifRQ message.

## 2.8.1 Usage Profile Table

Element   @Attribute	Num	Description/Contents
OTA_HotelAvailNotifRQ	1	Root element of the message.
@EchoToken	0..1	Optional
@TimeStamp	1	time of the transaction
@Version	1	Version is a mandatory attribute in OTA – therefore it must remain mandatory in HTNG in order to be able to use the same message
@TransactionIdentifier	0..1	Optional. If the availability update is the result of a reservation upload, the reservation number should be indicated in this field.
@MessageContentCode	1	The attribute refers to OTA code list MCC which includes RateAvail, RoomAvail, RoomRateAvail, SegmentAvail, SegmentRoomAvail, HouseAvail, HurdleRateUpdate. For hurdle rate update this should be set to 7.
OTA_HotelAvailNotifRQ / AvailStatusMessages	1	Must be sent for the message to have a meaning
@HotelCode	1	This is the code of the property whose availability is being updated
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage	0..n	Optional. Although all restrictions are optional, at least one should be sent for the message to have any meaning. Upper limit to be defined by trading partners.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / UniqueID	1	
@Type	1	Type refers to OTA code list UIT – nr 16 = Reference. This is used to identify each single availability status message for error reporting purposes.
@ID	1	ID is a unique incremental number for each availability message that identifies that specific message.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / StatusApplicationControl	1	Element defining application of controls being sent.



Element   @Attribute	Num	Description/Contents
@Start, @End	1	The first and last dates for which the availability update is being sent
@Mon, @Tue, @Weds, @Thur, @Fri, @Sat, @Sun	0..1	The day of the week indicators are used to communicate which days of the week the update pertains to. If one is sent they must all be sent.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / HurdleRate	1	
@Amount	1	This is the Hurdle rate the user wishes to set for the season sent.
@CurrencyCode	0..1	Currency of delta adjustment using ISO4217 codes. If the partner receiving the data does not support currency code for Hurdle rate, the currency code would be ignored.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / Delta	0..1	Incremental adjustment to the hurdle rate. Included if supported by trading partners.
@Amount	0..1	Amount of delta adjustment.
@CurrencyCode	0..1	Currency of delta adjustment using ISO4217 codes. If the partner receiving the data does not support currency code for Hurdle rate, the currency code would be ignored.
@Ceiling, @MaxSold	0..1	Limits on the application of the Delta
@InvTypeCode	0..1	This is the Room Type for which the Hurdle rate is sent (if sent on a room type by room type basis).
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay	0..1	If sending Hurdle Rate by Arrival Date and Length of Stay this element would be sent.
OTA_HotelAvailNotifRQ / AvailStatusMessages / AvailStatusMessage / LengthsOfStay / LengthOfStay	0..n	Conditionally mandatory. If LengthsOfStay is sent then there must be at least one LengthOfStay element.
@Time, @TimeUnit	0..1	Conditionally Required. If LengthsOfStay element included because sending Hurdle Rate by Arrival Date and Length of Stay then TimeUnit is set to 'Day' and Time is set to the corresponding Length of Stay; e.g. 1, 2, 3, etc.

### 2.8.2 Example Message

Scenario:

RMS “DEF” needs to transmit the following Hurdle rate update to CRS “123”

Hotel Code	HXCAIZZ
Date / Hurdle Rate:	Jan 1 <sup>st</sup> 2006 / €100.00
Date / Hurdle Rate:	Jan 2 <sup>nd</sup> 2006 / €123.00
Date / Hurdle Rate:	Jan 3 <sup>rd</sup> 2006 / €98.00

```
<OTA_HotelAvailNotifRQ xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA_HotelAvailNotifRQ.xsd" Version="1.001"
EchoToken="54638383" TransactionIdentifier="938389" Timestamp="2005-08-01T09:30:47-05:00"
MessageContentCode="7">
  <!-- MessageContentCode=7 for Hurdle Rate Update -->
  <AvailStatusMessages HotelCode="HXCAIZZ">
    <AvailStatusMessage>
      <StatusApplicationControl Start="2006-01-01" End="2006-01-01">
      </StatusApplicationControl>
      <HurdleRate Amount="100.00" CurrencyCode="EUR"/>
      <UniqueID Type="16" ID="1"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2006-01-02" End="2006-01-02">
      </StatusApplicationControl>
      <HurdleRate Amount="123.00" CurrencyCode="EUR"/>
      <UniqueID Type="16" ID="2"/>
    </AvailStatusMessage>
    <AvailStatusMessage>
      <StatusApplicationControl Start="2006-01-03" End="2006-01-03">
      </StatusApplicationControl>
      <HurdleRate Amount="98.00" CurrencyCode="EUR"/>
      <UniqueID Type="16" ID="3"/>
    </AvailStatusMessage>
  </AvailStatusMessages>
</OTA_HotelAvailNotifRQ>
```

```

    </AvailStatusMessages>
</OTA_HotelAvailNotifRQ>

```

## 2.9 Update Availability Response

The update response for any of the requests defined above is sent using the OTA\_HotelAvailNotifRS message.

### 2.9.1 Usage Profile Table

Element   @Attribute	Num	Description/Contents
OTA_HotelAvailNotifRQ	1	Root element of the message.
@EchoToken	0..1	Optional
@Version	1	Version is a mandatory attribute in OTA – therefore it must remain mandatory in HTNG in order to be able to use the same message
@TimeStamp	1	time of the transaction
@MessageContentCode	1	The attribute refers to OTA code list MCC which includes RateAvail, RoomAvail, RoomRateAvail, SegmentAvail, SegmentRoomAvail, HouseAvail, HurdleRateUpdate
OTA_HotelAvailNotifRS / Success	1	This is the annotation that the availability status message batch was received successfully. It could be combined with warning messages if some of the messages in the batch had issues
OTA_HotelAvailNotifRS / Warnings	0..1	Optional
OTA_HotelAvailNotifRS / Warnings / Warning	1..99	Mandatory
@Type	1	Refers to OTA EWT list (error warning type).
@Code	0..1	Optional . Refers to OTA list ERR . Should be used wherever possible
@RecordID	0..1	If the receiving system is able to identify within a batch of availability status messages which specific

Element   @Attribute	Num	Description/Contents
		message failed, the UniqueID of the message should be reported here.
OTA_HotelAvailNotifRS / Errors	0..1	Optional
OTA_HotelAvailNotifRS / Errors / Error	1..99	Mandatory
@Type	1	Mandatory in OTA. Refers to OTA EWT list (error warning type).
@Code	0..1	Optional . Refers to OTA list ERR . Should be used wherever possible
@RecordID	0..1	If the receiving system is able to identify within a batch of availability status messages which specific message failed, the UniqueID of the message should be reported here.

### 2.9.2 Example Message

#### Scenario:

Assume the example hurdle rate update request message from RMS “DEF” to CRS “123” shown in the previous section was re-sent the next day, on January 2<sup>nd</sup> 2006, due to some internal malfunction where the RMS did not mark the update as processed. This resulted in the first message trying to update the hurdle rate for a past date, January 01, 2006. Assume also that RMS “DEF” and CRS “123” have identified this error condition as a warning and agreed it should not stop processing but return a warning element so technical support at RMS “DEF” can look at the situation and ensure there are no major issues with the interface. In this case, the response message from CRS “123” to RMS “DEF” would look as shown below. Notice attribute RecordId in the warning element matches the value of UniqueId for the element causing the warning in the request message.

```
<OTA_HotelAvailNotifRS xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA_HotelAvailNotifRS.xsd" Version="1.001"
EchoToken="54638383" TransactionIdentifier="938389" Timestamp="2006-01-02T09:30:47-05:00">
<Success/>
<Warnings>
  <Warning Type="3" Code="458" RecordID="1">
    <!-- Type 3 => Business Rule, Code 458 => Date Outside Inventory Period -->
  </Warning>
</Warnings>
```

</OTA\_HotelAvailNotifRS>

## 2.10 FAQ

### *2.10.1 How should errors and warnings be used?*

The response messages have an option between sending a Success element with an optional collection of Warning elements or a collection of Error elements. It has been agreed that when the request message is not processed the response will only have error elements. When the message is processed then the Success element will be sent, along with any warnings indicating issues that did not prevent the processing but should result in some future correction by the implementers of the message. Every AvailStatusMessage element in the request will be assigned a unique identifier. In the cases where a AvailStatusMessage element caused the error or the warning, the value of attribute RecordId will be set to match the unique identifier to connect the two. The intent is for technical support from either party can use the additional information to identity the cause of the problem. It should be noted that there may be cases where the error or warning may be caused at a higher level than the AvailStatusMessage and the value of RecordId cannot be set.