



HTNG Point of Sale Specification Version 3.1

31 October 2014

About HTNG

Hotel Technology Next Generation (HTNG) is a non-profit association with a mission to foster, through collaboration and partnership, the development of next-generation systems and solutions that will enable hoteliers and their technology vendors to do business globally in the 21st century; to be recognized as a leading voice of the global hotel community, articulating the technology requirements of hotel companies of all sizes to the vendor community; and to facilitate the development of technology models for hospitality that will foster innovation, improve the guest experience, increase the effectiveness and efficiency of hotels, and create a healthy ecosystem of technology suppliers.

Copyright 2014, 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.

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. Visit <http://htng.org/ip-claims> to view third-party claims that have been disclosed to HTNG. HTNG offers no opinion as to whether claims listed on this site may apply to portions of this specification.

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 organization's members through the current terms and conditions of membership.

Table of contents

31 OCTOBER 2014	1
1 THIS SPECIFICATION AT A GLANCE.....	7
2 DOCUMENT INFORMATION	8
2.1 DOCUMENT HISTORY	8
2.2 DOCUMENT PURPOSE	9
2.3 SCOPE	9
2.4 RELATIONSHIP TO OTHER STANDARDS.....	10
2.5 USEFUL RESOURCES.....	10
2.6 AUDIENCE.....	10
2.7 OVERVIEW	10
2.8 KNOWN LIMITATIONS	11
3 COMPONENT SCENARIOS.....	12
3.1 ACCOUNT LOOKUP	12
3.1.1 <i>Overview</i>	12
3.1.2 <i>Roles</i>	12
3.1.3 <i>Use Case</i>	12
3.1.4 <i>Message Flows</i>	13
3.1.5 <i>Sample Request</i>	13
3.1.6 <i>Sample Response</i>	13
3.2 CHARGE POSTING	14
3.2.1 <i>Overview</i>	14
3.2.2 <i>Roles</i>	15
3.2.3 <i>Use Case</i>	15
3.2.4 <i>Message Flows</i>	16
3.2.5 <i>Sample Request – Standard Charge Posting</i>	16
3.2.6 <i>Sample Response</i>	17
3.3 VOID POSTING.....	18
3.3.1 <i>Overview</i>	18
3.3.2 <i>Roles</i>	18
3.3.3 <i>Use Case</i>	18
3.3.4 <i>Message Flows</i>	19
3.3.5 <i>Sample Request</i>	19
3.3.6 <i>Sample Response</i>	20
3.4 CHECK DETAIL LOOKUP	20
3.4.1 <i>Overview</i>	20
3.4.2 <i>Roles</i>	20
3.4.3 <i>Use Case</i>	21
3.4.4 <i>Message Flows</i>	22

3.4.5	<i>Sample Request</i>	22
3.4.6	<i>Sample Response</i>	22
3.5	SYNCHRONIZE CACHE – REAL–TIME	22
3.5.1	<i>Overview</i>	22
3.5.2	<i>Roles</i>	23
3.5.3	<i>Use Case</i>	23
3.5.4	<i>Message Flows</i>	24
3.5.5	<i>Sample Request</i>	24
3.5.6	<i>Sample Response</i>	25
3.6	SYNCHRONIZE CACHE – PUSH.....	25
3.6.1	<i>Overview</i>	25
3.6.2	<i>Roles</i>	25
3.6.3	<i>Use Case</i>	25
3.6.4	<i>Message Flows</i>	26
3.6.5	<i>Sample Request</i>	26
3.6.6	<i>Sample Response</i>	27
3.7	SYNCHRONIZE CACHE – PULL	27
3.7.1	<i>Overview</i>	27
3.7.2	<i>Roles</i>	28
3.7.3	<i>Use Case</i>	28
3.7.4	<i>Message Flows</i>	29
3.7.5	<i>Sample Request</i>	29
3.7.6	<i>Sample Response</i>	29
3.8	OFFLINE STATUS NOTIFICATION	30
3.8.1	<i>Overview</i>	30
3.8.2	<i>Roles</i>	30
3.8.3	<i>Use Case</i>	30
3.8.4	<i>Message Flows</i>	31
3.8.5	<i>Sample Request</i>	31
3.8.6	<i>Sample Response</i>	31
3.9	FINANCIAL RECONCILIATION	32
3.9.1	<i>Overview</i>	32
3.9.2	<i>Roles</i>	32
3.9.3	<i>Use Case</i>	32
3.9.4	<i>Message Flows</i>	33
3.9.5	<i>Sample Request</i>	33
3.9.6	<i>Sample Response</i>	34
3.10	RAPID LOYALTY REDEMPTION	34
3.10.1	<i>Overview</i>	34
3.10.2	<i>Roles</i>	34
3.10.3	<i>Use Case</i>	35
3.10.4	<i>Message Flows</i>	35

3.10.5	Sample Request.....	35
3.10.6	Sample Response	36
3.10.7	Sample Request.....	36
3.10.8	Sample Response	37
3.11	LOYALTY VOUCHER REDEMPTION	37
3.11.1	Overview	37
3.11.2	Roles.....	37
3.11.3	Use Case	37
3.11.4	Message Flows	38
3.11.5	Sample Request.....	38
3.11.6	Sample Response	39
3.12	LOYALTY BALANCE INQUIRY	39
3.12.1	Overview	39
3.12.2	Roles.....	39
3.12.3	Use Case	39
3.12.4	Message Flows	40
3.12.5	Sample Request.....	40
3.12.6	Sample Response	41
3.13	ASSOCIATING TRANSACTIONS WITH LOYALTY NUMBER	42
4	MESSAGES.....	44
4.1	ACCOUNT LOOKUP / SYNCHRONIZE CACHE – PULL	44
4.1.1	Data Element Table – Request.....	44
4.1.2	Data Element Table – Response.....	46
4.2	CHARGE / VOID POSTING / FINANCIAL RECONCILIATION / RAPID LOYALTY REDEMPTION / LOYALTY VOUCHER REDEMPTION	53
4.2.1	Data Element Table – Request	53
4.2.2	Data Element Table – Response.....	59
4.3	CHECK DETAIL LOOKUP	62
4.3.1	Data Element Table – Request	62
4.3.2	Data Element Table – Response.....	63
4.4	SYNCHRONIZE CACHE – REAL-TIME/ SYNCHRONIZE CACHE – PUSH.....	65
4.4.1	Data Element Table – Request	65
4.4.2	Data Element Table – Response.....	72
4.5	OFFLINE STATUS NOTIFICATION	73
4.5.1	Data Element Table – Request	73
4.5.2	Data Element Table – Response.....	74
5	APPENDICES	76
5.1	TERMS	76
5.2	IMPLEMENTATION NOTES.....	77
5.2.1	Void versus Refund	77

5.2.2	<i>Financial Reconciliation</i>	77
5.2.3	<i>Account Posting Payload</i>	78
5.2.4	<i>Summary Transaction Posting</i>	79
5.2.5	<i>Best Practices for Data Mining</i>	80
5.3	REFERENCED DOCUMENTS	81

1 This Specification at a Glance

This specification outlines a common data definition and protocol implementation for the venerable and commonly implemented hospitality industry “Property Management System (PMS) interface.” Compared to legacy interface communication methods, this interface increases scalability and speed of transactions, reduces redundancy and improves suitability for implementation within a wide area network or cloud computing environment. The common data definition is designed to provide backwards compatibility so that all data included in vendor-specific interfaces can be transmitted using this interface specification.

This specification outlines the data structures and methods used by PMS and Point of Sale (POS) systems to be able to:

- Look up and verify guest or membership account information
- Post charges to a guest or membership account
- Allow front desk or membership accounting personnel to look up charge posting details in the same format as the transaction was originally provided to a guest/member to help resolve billing questions
- Facilitate offline account lookup and charge posting features
- Automate an end-of-day audit of interface transaction postings aimed to ensure the accuracy of financial data posted via the interface and to serve as an implementation testing tool
- Synchronize data between PMS and POS systems for offline functionality
- Allow for the posting of all tender types and accounts, in detail or in summary, that have been configured in the interface
- Translate value between points and merchandise dollar value and handle the necessary accounting for the purchase transaction
- Allow guests to use loyalty points and/or vouchers, in lieu of or in conjunction with cash payments for goods or services
- Enable loyalty balance inquiries from the POS (or other) system and enable loyalty accounts to be associated with transactions to accurately track purchases made by the loyalty program member

The data definitions are built upon elements of the [HTNG Customer Profile v3.0](#) specification and the [HTNG Folio Detail Exchange v2.0](#) specification. This includes the lookup of customer profiles and the representation of transactional information. Similar to all recent HTNG efforts, these messages share OpenTravel Alliance message components.

2 Document Information

2.1 Document History

Version	Date	Author	Comments
1.0	19 Oct 2012	Point of Sale (POS) Workgroup	Released spec – addressed Account Lookup, Charge Posting, Void Posting, Check Detail Lookup use cases
1.01	22 Oct 2012	POS Workgroup	Added/updated offline use cases
1.02	29 Oct 2012	POS Workgroup	Updated use cases and Sections 1 & 2
1.03– 1.09	5 Nov 2012 – 17 Dec 2012	POS Workgroup	Updated use cases
1.10– 1.12	02 Jan 2013 – 21 Jan 2013	POS Workgroup	Updated sample messages
1.13 – 1.18	28 Jan 2013 – 18 March 2013	POS Workgroup	Edited sample messages, data element tables
1.90	21 Mar 2013	Kylene Reese	Prepared for member review period
1.95	09 Apr 2013	Kylene Reese	Prepared spec for workgroup vote
2.0	19 Apr 2013	POS Workgroup	Released spec – includes addition of offline functionality and end-of-day reconciliation
2.01	14 Jan 2014	POS Workgroup	Added loyalty-related use cases
2.02	27 Jan 2014	POS Workgroup	Updated scenarios
2.03– .04	Feb 2014	POS Workgroup	Completed data element identification
2.90	20 Mar 2014	Kylene Reese	Prepared spec for member review period
2.99	09 Apr 2014	Kylene Reese	Prepared spec for workgroup vote
3.0	18 April 2014	POS Workgroup	Publicly released spec, which incorporates loyalty scenarios
3.01	07 Jul 2014	Jay Rosamilia	Fixed namespaces in sample messages

2.2 Document Purpose

A myriad of different Point of Sale (POS) systems post charges to Property Management Systems (PMS) using different data formats and a mix of communication methods, and provide varying levels of transaction detail. The wide range of interfaces with different features complicates and increases the cost of supporting multiple interfaces to a single system and reduces the number of consumer choices available for implementation.

Many legacy systems use serial communications with limited data speeds that can limit transaction speed or cause communication bottlenecks in businesses with higher transaction volumes. This can also happen at peak hours and can be more difficult to implement and manage over a wide area network due to specific cabling requirements, distance limitations or need for protocol conversion equipment.

Although serial interfaces are inherently secure due to their point-to-point known connectivity, web services offer a secure means of transport and more flexibility for implementation in cloud computing or WAN environments and provide the ability to use the same network infrastructure for multiple interfaces.

This specification provides the data structure and payload that all systems can use to look up accounts, post room charges and other financial transactions in the level of detail required for the business and agreed upon by the parties implementing the interface. In doing this, compatibility with all systems is ensured while still allowing for flexibility in the data that is transmitted to a given PMS or back-of-house system.

By using a web services interface, the communication methods and protocols are unified, easing the system support and allowing systems to be implemented across a range of network topologies without the use of special cabling and protocol conversion equipment.

2.3 Scope

This document includes use cases of key transactions, including account lookup, posting, voids, refunds, check detail lookup, offline transaction handling, end-of-day financial reconciliation and loyalty functionality, including rapid loyalty redemption, voucher redemption and loyalty balance inquiry.

The data storage format and data management mechanisms of account information looked up via this interface is left to the developer.

No information or guidance is given related to the configuration and setup that determines how posted transactions are further processed for use in accounting, membership or other systems

that might receive (or have access to) the PMS. This is left to business partners implementing this interface or to other interface specifications entirely.

2.4 Relationship to Other Standards

This specification and its supporting schemas leverage the existing OpenTravel Alliance methodology for message construction and draw upon data definitions common to several HTNG specifications as of April 2014.

Related specifications:

- [HTNG Customer Profile v3.0](#)
- [HTNG Folio Detail Exchange v2.0](#)
- [HTNG Guest & Room Status Messaging v2.0](#)
- [HTNG Single Guest Itinerary v2.0](#)
- [Open Travel Alliance Specifications](#)

2.5 Useful Resources

- [Implementing Web Services Using HTNG Specifications – A Quick Start Guide for Software Developers](#)
- HTNG Discussion Board – currently available at <http://www2.htng.org/discussion>

2.6 Audience

This document is intended to be used by software developers, integrators and operators who are designing and implementing interfaces between hospitality PMS, POS, Membership, Accounting and other systems that post to a PMS.

2.7 Overview

Content contained in this document has been designed to provide a firm understanding of the components that must be implemented, or can optionally be implemented, to meet the requirements of this specification. A brief overview of each section is as follows:

Section 3 – Component Scenarios

This section includes definitions for each component used to implement the interface. Each component definition includes an explanation of system roles and provides a use case, message flow diagram and sample message data for the component.

Section 4 – Messages

Detailed Data Element tables and sample messages are provided in this section.

Section 5 – Appendices

This section includes implementation requirements, links, any referenced documents, as well as common HTNG schema components referenced in this document. The technical artifacts (XSDs and WSDLs) for this certification release can be found in a separate ZIP file included with the specification.

2.8 Known Limitations

The data configuration, pre- and post-processing of data transmitted via the interface, and coordination of configuration for a specific interface between the POS and PMS is left to each software developer, respectively.

3 Component Scenarios

Note: Loyalty programs may work on a point system or currency and oftentimes, there is a need to convert between the two. Trading partners will need to decide this. If any conversion between points and currency is needed, it is assumed that the Customer Loyalty Responder will handle the conversion.

3.1 Account Lookup

3.1.1 Overview

This use case describes the business and system processes and requirements for the search for accounts (for example, guest folios, membership accounts, house accounts, AR accounts) along with the associated profile. This may be a system-to-system interaction or it may be a system interacting with a locally cached copy of this data (see Synchronize Cache use cases below).

3.1.2 Roles

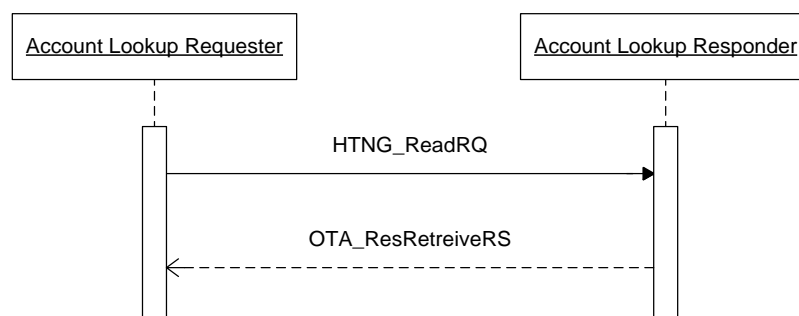
Role Name	Definition	Example
Account Lookup Requester	A system that needs to determine whether a particular guest is staying in the hotel, or to look up an existing (house, membership, etc.) account.	Point of Sale
Account Lookup Responder	A system that holds information about hotel guests, their associated reservations, and accounts.	<ul style="list-style-type: none">Property Management System (PMS)Cache Holder

3.1.3 Use Case

Assumption:	The Account Lookup Requester and Account Lookup Responder have an agreed set of parameters that can be used to identify an account.
Pre-condition:	The Account Lookup Requester is aware of some basic information that can be used to identify a specific account (e.g., Surname, Guest ID, Membership ID, Guest Folio ID, etc.).
Trigger:	The Account Lookup Requester has the need to uniquely identify an account.
Basic Course of Events:	<ol style="list-style-type: none">1) The Account Lookup Requester sends a request to the Account Lookup Responder containing one or more parameters.2) The Account Lookup Responder searches the database and locates matching accounts and their associated profiles.

	<p>3) The Account Lookup Responder builds a message containing the matching results.</p> <p>4) The Account Lookup Responder returns the message to the Account Lookup Requester.</p>
Post-condition:	The Account Lookup Requester can then display the results to the screen, link an account to a check, or close a check to the guest account.
Exception Path:	The Account Lookup Responder is unable to locate an account using the criteria supplied by the Account Lookup Requester.
Alternative Paths:	None

3.1.4 Message Flows



3.1.5 Sample Request

```

<HTNG_ReadRQ EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
  xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <POS>
    <ota:Source>
      <ota:RequestorID Type="0" ID_Context="a" ID="a"/>
    </ota:Source>
  </POS>
  <UniqueID Type="0" ID_Context="a" ID="a"/>
  <ReadRequests>
    <HotelReadRequest HotelCode="a" ChainCode="a" BrandCode="a">
      <Verification>
        <ota:PersonName PartialName="true">
          <ota:GivenName>a</ota:GivenName>
          <ota:Surname>a</ota:Surname>
        </ota:PersonName>
        <ota:CustLoyalty MembershipID="798654231" ProgramID="HTNG_REWARDS" LoyalLevel="Gold"/>
        <ota:ota:ReservationTimeSpan End="1967-08-13" Start="1967-08-13"/>
        <Room RoomID="101"/>
      </Verification>
    </HotelReadRequest>
  </ReadRequests>
</HTNG_ReadRQ>
  
```

3.1.6 Sample Response

```

<OTA_ResRetrieveRS EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
  xmlns="http://www.opentravel.org/OTA/2003/05">
  <Success/>
  <Warnings>
    <Warning Type="0" Status="a" RecordID="a" ShortText="a" Code="0">String</Warning>
  </Warnings>
  <ReservationsList>
  
```

```

<HotelReservation RoomStayReservation="true" ResStatus="Reserved|In-house|Checked
out|Cancelled">
  <RoomStays>
    <RoomStay MarketCode="T" DiscountCode="DSC123" RoomStayStatus="In-house">
      <RoomRates>
        <RoomRate RoomTypeCode="KING" RatePlanCode="PKG123" RoomID="101">
          <GuestCounts>
            <GuestCount AgeQualifyingCode="10" Count="2"/>
            <GuestCount AgeQualifyingCode="8" Count="2"/>
          </GuestCounts>
        </RoomRate>
      </RoomRates>
    </RoomStay>
    <TimeSpan Start="2012-08-20" End="2012-08-21"/>
  </RoomStays>
  <ResGuests>
    <ResGuest VIP="true" PrimaryIndicator="true">
      <Profiles>
        <ProfileInfo>
          <UniqueID Type="" ID="" ID_Context=""/>
        <Profile>
          <Customer>
            <PersonName>
              <NamePrefix>Mr. </NamePrefix>
              <GivenName>Jay</GivenName>
              <Surname>Rosamili</Surname>
            </PersonName>
            <CustLoyalty MembershipID="798654231" ProgramID="HTNG_REWARDS"
Loyal Level="Gold"/>
          </Customer>
        </Profile>
      </ProfileInfo>
    </Profiles>
    <Comments>
      <Comment Name="" GuestVisible="false">
        <Text Language="en-us"/>
      </Comment>
    </Comments>
  </ResGuest>
</ResGuests>
<TPA_Extensions>
  <TPA_Extension>
    <AvailableCredit CurrencyCode="USD" Amount="128.37" DecimalPlaces="0"/>
  </TPA_Extension>
</TPA_Extensions>
</HotelReservation>
</ReservationsList>
</OTA_ResRetrieveRS>

```

3.2 Charge Posting

3.2.1 Overview

This use case describes the business and system processes and requirements for the posting of charges to previously identified accounts (for example guest folios, membership accounts, house accounts, AR accounts).

Note that the message payload may be a mixture of positive and negative values. This is particularly useful when performing merchandise exchanges where the value of the originally purchased item differs from value of the new item.

If systems are unable to communicate due to an offline state or network failure, the sending system should queue this transaction for later delivery. It should be noted, however, that these transactions may have been manually posted and therefore, the receiving system should ensure it has the appropriate logic to avoid duplicate postings and/or a method to identify each unique posting method (i.e. a manual posting vs. an automated posting via this interface).

3.2.2 Roles

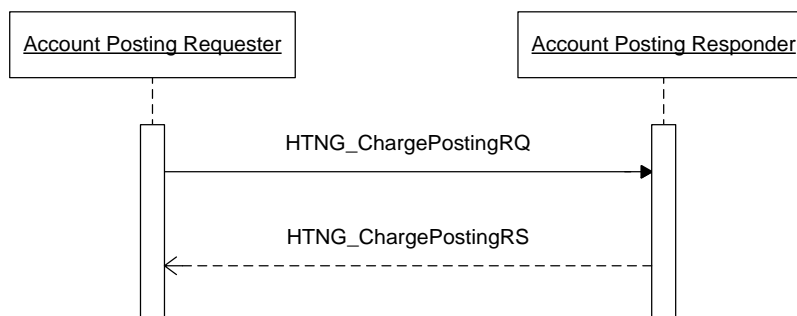
Role Name	Definition	Example
Account Posting Requester	A system that has the need to post charges to guest rooms or accounts.	Charge Posting System
Account Posting Responder	A system that houses guest and account information; receives charge requests and posts them to the appropriate folios.	<ul style="list-style-type: none">• Folio/Membership Accounting System• PMS

3.2.3 Use Case

Assumptions:	<ul style="list-style-type: none">• The Account Posting Requester and Account Posting Responder have an agreed set of parameters that can be used to identify an account.• Determining whether a guest has sufficient credit will be based on the implementation.
Pre-conditions:	<ul style="list-style-type: none">• The Account to which the charges will be posted has been previously identified.• The credits balance with the debits in each message.
Trigger:	The Account Posting Requester has the need to post a charge to an account.
Basic Course of Events:	<ol style="list-style-type: none">1) The Account Posting Requester sends a request to the Account Posting Responder containing one or more parameters used to identify the account along with the supporting information representing the charge(s) to be posted.2) The Account Posting Responder searches the database and locates matching account and its associated profile.3) The Account Posting Responder posts the charge(s) to the appropriate account.4) The Account Posting Responder returns the response message to the Account Posting Requester.
Post-condition:	The Account Posting Requester posts the appropriate payment to the check.

Exception Path:	<ul style="list-style-type: none"> The Account Lookup Responder transmits a negative acknowledgement message in response to the account charge message. <ul style="list-style-type: none"> The Account Posting Responder is unable to locate an account using the criteria supplied by the Account Lookup Requester. The Account Posting Responder determines the account is not in good standing or otherwise not eligible (based upon the Account Posting Responder's business rules) for charge posting (i.e., Credit Limit, In-house guest, etc.).
Alternative Paths:	None

3.2.4 Message Flows



3.2.5 Sample Request – Standard Charge Posting

```

<HTNG_ChargePostingRQ EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <POS>
    <ota:Source>
      <ota:RequestorID Type="0" ID_Context="a" ID="a">
        <ota:CompanyName CodeContext="a" CompanyShortName="a" Department="a" Division="a"
Code="a"/>
      </ota:RequestorID>
    </ota:Source>
  </POS>
  <PropertyInfo ChainCode="a" HotelName="a" BrandCode="a" HotelCode="a" BrandName="a"
HotelCodeContext="a"/>
  <Posting ID="67436984" ZoomInKey="" Void="false" BusinessPeriod="2012-02-18">
    <RevenueCenter ID="3" Description="">
      <Terminal ID="">
    </RevenueCenter>
    <TotalPostingAmount CurrencyCode="AAA" Amount="66.00" DecimalPlaces="0"/>
    <ota:ServerInfo EmployeeID="3215"/>
    <ota:CashierInfo EmployeeID="3241"/>
    <Transaction TicketID="4651" Table="23" Covers="2" OpenTime="2012-12-31T06:23"
CloseTime="2012-12-31T06:52" MealPeriodID="" DigitsDialled="" Duration="" Extension="">
      <RevenueDetails>
        <RevenueDetail ReferenceID="" CurrencyCode="AAA" Amount="20.00" Description="a"
DecimalPlaces="0" RevenueCategoryCode="14" SubTypeID="Local">
          <FolioIDs>
            <FolioID>0</FolioID>
          </FolioIDs>
          <ExtendedPrice Type="0" AmountBeforeTax="20.00" CurrencyCode="USD" Quantity="2"
DecimalPlaces="0"/>
        </RevenueDetail>
        <RevenueDetail PMSRevenueCode="" ReferenceID="" CurrencyCode="USD" Amount="20.000"
Description="Tip" DecimalPlaces="0" RevenueCategoryCode="10" SubTypeID="TIP">
  
```

```
<FolioIDs>
  <FolioID>0</FolioID>
</FolioIDs>
<ExtendedPrice Type="0" AmountBeforeTax="20.00" CurrencyCode="USD" Quantity="2"
DecimalPlaces="0"/>
</RevenueDetail>
<RevenueDetail ReferenceID="" CurrencyCode="USD" Amount="20.000" Description="Tip"
DecimalPlaces="0" RevenueCategoryCode="10" SubTypeID="SERVICECHARGE">
  <FolioIDs>
    <FolioID>0</FolioID>
  </FolioIDs>
  <ExtendedPrice Type="0" AmountBeforeTax="20.00" CurrencyCode="USD" Quantity="2"
DecimalPlaces="0"/>
</RevenueDetail>
</RevenueDetails>
<ota:TaxItems CurrencyCode="AAA" Amount="6.00" DecimalPlaces="0">
  <ota:Tax CurrencyCode="AAA" Amount="6.00" DecimalPlaces="0" Code="0">
    <ota:TaxDescription Name="a"/>
  </ota:Tax>
</ota:TaxItems>
<Tenders>
  <RevenueDetail ReferenceID="" CurrencyCode="USD" Amount="66.00" Description="Room Charge"
DecimalPlaces="0" RevenueCategoryCode="6" SubTypeID="RMCHG" TenderID="19">
    <FolioIDs>
      <FolioID>0</FolioID>
    </FolioIDs>
    <Account Type="Account|Room|Extension" ID="123456"/>
  </RevenueDetail>
</Tenders>
</Transaction>
</Posting>
</HTNG_ChargePostingRQ>
```

3.2.6 Sample Response

```
<HTNG_ChargePostingRS EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <Success/>
  <PostingGUID>String</PostingGUID>
  <TotalPostingAmount CurrencyCode="AAA" Amount="66.00" DecimalPlaces="0"/>
  <Accounts>
    <Account Type="0" ID_Context="a" ID="a">
      <PersonName>
        <GivenName/>
        <Surname/>
      </PersonName>
    </Account>
  </Accounts>
</HTNG_ChargePostingRS>
```

3.3 Void Posting

3.3.1 Overview

This use case describes the business and system processes and requirements for voiding previously posted transaction(s) on accounts (for example guest folios, membership accounts, house accounts, AR accounts). This includes a full audit reversal which can be used by the receiving system to properly handle package breakage or routing concerns.

3.3.2 Roles

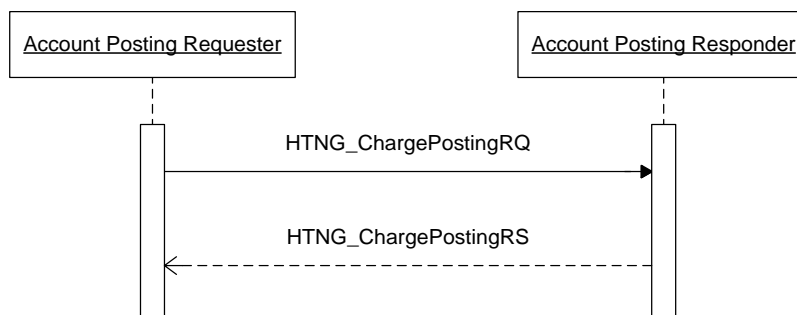
Role Name	Definition	Example
Account Posting Requester	A system that has the need to post charges to guest rooms or accounts.	Charge Posting System
Account Posting Responder	A system that houses guest and account information; receives charge requests and posts them to the appropriate folios.	<ul style="list-style-type: none">Folio/Membership Accounting SystemPMS

3.3.3 Use Case

Assumption:	The Account Posting Requester and Account Posting Responder have an agreed set of parameters that can be used to identify an account and determine credit limits for an account.
Pre-conditions:	<ul style="list-style-type: none">The transaction to be voided has been identified by the Account Posting Requester.The Account to which the charges should be voided has been previously identified by the Account Posting Requester.
Trigger:	The Account Posting Requester has the need to void a charge previously made to an account.

Basic Course of Events:	<ol style="list-style-type: none"> 1) The Account Posting Requester sends a request to the Account Posting Responder containing one or more parameters used to identify the transaction AND the account along with the supporting information representing the void to be posted. 2) The Account Posting Responder searches the database and locates matching transaction. 3) The Account Posting Responder searches the database and locates matching account and its associated profile. 4) The Account Posting Responder voids the appropriate transaction and any necessary backing detail to the account and applies a specific transaction identifier. 5) The Account Posting Responder returns the response message to the Account Posting Requester.
Post-condition:	The Account Posting Requester may now re-apply payment for the charge as necessary.
Exception Paths:	<ul style="list-style-type: none"> • The Account Posting Responder is unable to locate a transaction or an account using the criteria supplied by the Account Lookup Requester. • The Account Posting Responder determines the transaction is not eligible based upon the Account Posting Responder's business rules for voiding charges. • The Account Lookup Responder transmits a negative acknowledgement message in response to the void charge message.
Alternative Paths:	None

3.3.4 Message Flows



3.3.5 Sample Request

```

<HTNG_ChargePostingRQ EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
  xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <POS>
    <ota:Source>
      <ota:RequestorID Type="0" ID_Context="a" ID="a">

```

```

    <ota:CompanyName CompanyShortName="a" Department="a" Division="a"/>
  </ota:RequestorID>
</ota:Source>
</POS>
<UniqueID Type="0" ID_Context="a" ID="a"/>
<PropertyInfo ChainCode="a" HotelName="a" BrandCode="a" HotelCode="a" BrandName="a"
HotelCodeContext="a"/>
<Posting ID=" " ZoomInKey=" " Void="true">
  <RevenueCenter ID="3" Description=" ">
    <Terminal ID=" "/>
  </RevenueCenter>
  <TotalPostingAmount CurrencyCode="AAA" Amount="- 5. 00" DecimalPlaces="1"/>
  <ota:ServerInfo EmployeeId="3215"/>
  <ota:CasherInfo EmployeeId="3241"/>
  <Transaction TicketID="4651" Table="23" Covers="2" OpenTime="2012-12-31T06:23"
CloseTime="2012-12-31T06:52" MealPeriodID=" " DigitsDialed=" " Duration=" " Extension=" ">
    <RevenueDetails>
      <RevenueDetail ReferenceID="Sort Order?" CurrencyCode="AAA" Amount="10.000"
Description="T Shirt" DecimalPlaces="1" RevenueCategoryCode="14" SubTypeID="Local">
        <ExtendedPrice Type="0" AmountBeforeTax="20.00" CurrencyCode="USD" Quantity="2"
DecimalPlaces="1"/>
      </RevenueDetail>
      <RevenueDetail ReferenceID="Sort Order?" CurrencyCode="AAA" Amount="- 15.000"
Description="Sweatshirt" DecimalPlaces="1" RevenueCategoryCode="14" SubTypeID="Local">
        <ExtendedPrice Type="0" AmountBeforeTax="20.00" CurrencyCode="USD" Quantity="2"
DecimalPlaces="1"/>
      </RevenueDetail>
    </RevenueDetails>
    <ota:TaxItems CurrencyCode="AAA" Amount="1.123" DecimalPlaces="1">
      <ota:Tax CurrencyCode="AAA" Amount="1.123" DecimalPlaces="1" Code="0">
        <ota:TaxDescription Name="a"/>
      </ota:Tax>
    </ota:TaxItems>
    <Tenders>
      <RevenueDetail ReferenceID=" " CurrencyCode="USD" Amount="- 5.000" Description="Room
Charge" DecimalPlaces="2" RevenueCategoryCode="6" SubTypeID="RMCHG" TenderID="23"/>
    </Tenders>
  </Transaction>
</Posting>
</HTNG_ChargePostingRQ>

```

3.3.6 Sample Response

```

<HTNG_ChargePostingRS EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <Success/>
  <PostingGUID>String</PostingGUID>
  <TotalPostingAmount CurrencyCode="AAA" Amount="1.123" DecimalPlaces="1"/>
  <Accounts>
    <Account Type="0" ID="a"/>
  </Accounts>
</HTNG_ChargePostingRS>

```

3.4 Check Detail Lookup

3.4.1 Overview

This use case describes the business and system processes and requirements for the search for Point of Sale check details.

3.4.2 Roles

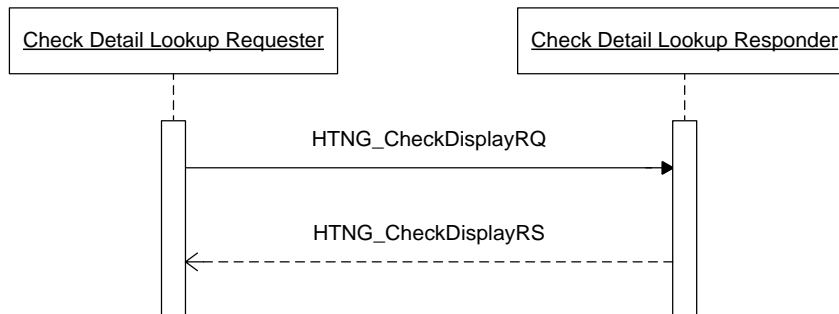
Role Name	Definition	Example
Check Detail Lookup Requester	A system that has previously received a charge posting but would like to view the charge detail.	<ul style="list-style-type: none"> Folio/Membership Accounting System PMS

Check Detail Lookup Responder	A system that had previously posted a charge to another system and is able to present a full representation of the original charge.	Charge Posting System
-------------------------------	---	-----------------------

3.4.3 Use Case

Assumption:	The Check Detail Lookup Requester and Check Detail Lookup Responder have an agreed set of parameters that can be used to identify a Point of Sale check.
Pre-condition:	The Check Detail Lookup Requester is aware of some basic information that can be used to identify a specific Point of Sale check (e.g., Check Number).
Trigger:	The Check Detail Lookup Requester has the need to uniquely identify a Point of Sale check.
Basic Course of Events:	<ol style="list-style-type: none">1) The Check Detail Lookup Requester sends a request to the Check Detail Lookup Responder containing one or more parameters.2) The Check Detail Lookup Responder searches the database and locates Point of Sale check details.3) The Check Detail Lookup Responder builds a message containing the matching results.4) The Check Detail Lookup Responder returns the message to the Check Detail Lookup Requester.
Post-condition:	The Check Detail Lookup Requester has the Point of Sale check detail in context and can display the results to the screen.
Exception Path:	The Check Detail Lookup Responder is unable to locate Point of Sale check details using the criteria supplied by the Check Detail Lookup Requester.
Alternative Path:	None

3.4.4 Message Flows



3.4.5 Sample Request

```

<HTNG_CheckDisplayRQ EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
xml ns="http://htng.org/2014B" xml ns:ota="http://www.opentravel.org/OTA/2003/05">
  <POS>
    <ota:Source>
      <ota:RequestorID Type="0" ID_Context="a" ID="a">
        <ota:CompanyName CompanyShortName="a" Department="a" Division="a"/>
      </ota:RequestorID>
    </ota:Source>
  </POS>
  <UniqueID Type="0" ID_Context="a" ID="a"/>
  <PropertyInfo ChainCode="a" HotelName="a" BrandCode="a" HotelCode="a"/>
  <CheckZoom ZoomInKey="SOME-GUID" TextFormat="PlainText"/>
</HTNG_CheckDisplayRQ>
  
```

3.4.6 Sample Response

```

<HTNG_CheckDisplayRS EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
xml ns="http://htng.org/2014B" xml ns:ota="http://www.opentravel.org/OTA/2003/05">
  <Success/>
  <CheckZoom ZoomInKey="SOME-GUID" TextFormat="PlainText">
    Restaurant
    16 Laura T
    -----
    Tbl 712/1    Chk 1394    Gst 15
        04/13/12 05:55:24
    -----
    15 MEPS BUFFET          142.50
        %Gratuity           25.65
        Tax:                 11.40
    MEPS FOOD0044245
        Room Charge         179.55
    ----- 16 Check Closed -----
    ----- 04/13/12 05:55:46 -----
  </CheckZoom>
</HTNG_CheckDisplayRS>
  
```

3.5 Synchronize Cache – Real-time

3.5.1 Overview

This use case describes the business and system processes and requirements for the real-time notification of changes to in-house reservations, house accounts, membership accounts, etc. and their associated profiles to systems leveraging a cache.

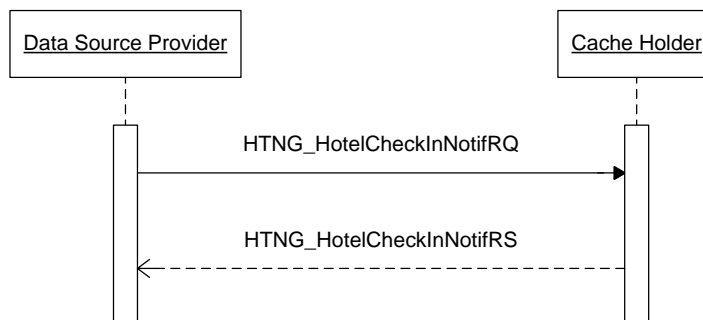
3.5.2 Roles

Role	Description	Examples
Cache Holder	A system that caches a view of (current, arriving, or departing) hotel guests and accounts for the purpose of being able to perform local lookups in case communications between systems becomes unavailable.	<ul style="list-style-type: none">• Retail Point of Sale (POS) System• Restaurant POS System• F&B POS System• Spa POS System
Data Source Provider	A system that publishes changes to its data to another system.	Property Management System

3.5.3 Use Case

Assumptions:	<ul style="list-style-type: none">• The HTNG Protocol & Message Transport Event Notification specification has been implemented.• The Cache Holder provides the data source provider with a URL where the updates should be posted.• The Data Source Provider accepts the subscription request and provides a unique subscription identifier to the Cache Holder.• The Data Source Provider notifies the Cache Holder of any record updates on an ongoing basis until the subscription is cancelled.
Pre-condition:	None
Trigger:	A change has been made to an in-house account or its associated profile.
Basic Course of Events:	<ol style="list-style-type: none">1) The Data Source Provider gathers the appropriate data values corresponding to the account and the associated profile.2) The Data Source Provider constructs the appropriate message and transmits it to the Cache Holder.
Post-condition:	The Cache Holder receives the message, parses its content and updates its local cache with the appropriate values.
Exception Path:	The Data Source Provider is unable to push an update to the Cache Holder and must queue the message for later delivery.
Alternative Paths:	None

3.5.4 Message Flows



3.5.5 Sample Request

```

<HTNG_HotelCheckInNotifRQ EchoToken="33f07f05-84e5-48ff-b772-894119d5c194" TimeStamp="2011-08-24T09:30:47Z" Version="2.0" Target="Production" xmlns="http://htng.org/2014B"
xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <PropertyInfo ChainCode="STARWOOD" HotelName="Westin San Diego" BrandCode="WESTIN"
Hotel Code="2134" Hotel CodeContext="CRES" />
  <AffectedGuests>
    <UniqueID Type="1" ID="GST123" />
    <UniqueID Type="14" ID="RES123" />
  </AffectedGuests>
  <Room RoomID="STE100">
    <RoomType RoomID="STE100" InvBlockCode="GRP123" />
    <TelephoneExtensions>
      <TelephoneExtension>71200</TelephoneExtension>
    </TelephoneExtensions>
  </Room>
  <HotelReservations>
    <ota:HotelReservation ResStatus="Checked In" RoomStayReservation="true">
      <ota:UniqueID Type="14" ID="RES123" />
      <ota:RoomStays>
        <ota:RoomStay MarketCode="T" DiscountCode="DSC123" RoomStayStatus="In-house">
          <RoomRates>
            <RoomRate RoomTypeCode="KING" RatePlanCode="PKG123">
              <GuestCounts>
                <GuestCount AgeQualifyingCode="10" Count="2" />
                <GuestCount AgeQualifyingCode="8" Count="2" />
              </GuestCounts>
            </RoomRate>
          </RoomRates>
          <ota:TimeSpan End="2011-09-18" Start="2011-09-20" />
        </ota:RoomStay>
      </ota:RoomStays>
      <ota:ResGuests>
        <ResGuest VIP="true" PrimaryIndicator="true">
          <Profiles>
            <ProfileInfo>
              <UniqueID Type="" ID="" />
              <Profile>
                <Customer>
                  <PersonName>
                    <NamePrefix>Mr. </NamePrefix>
                    <GivenName>Jay</GivenName>
                    <Surname>Rosamilia</Surname>
                  </PersonName>
                  <CustLoyalty MembershipID="798654231" ProgramID="HTNG_REWARDS"
Loyal Level="Gold" />
                </Customer>
              </Profile>
            </ProfileInfo>
          </Profiles>
          <Comments>
            <Comment Name="" GuestViable="false">
              <Text Language="en-us" />
            </Comment>
          </Comments>
        </ResGuest>
      </ota:ResGuests>
    </ota:HotelReservation>
  </HotelReservations>
</HTNG_HotelCheckInNotifRQ>
  
```

```

    </Comments>
  </ResGuest>
</ota:ResGuests>
<TPA_Extensions>
  <TPA_Extension>
    <AvailableCredit CurrencyCode="USD" Amount="128.37" DecimalPlaces="0"/>
  </TPA_Extension>
</TPA_Extensions>
</ota:HotelReservation>
</HotelReservations>
</HTNG_HotelCheckInNotifRS>

```

3.5.6 Sample Response

```

<HTNG_HotelCheckInNotifRS EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0"
Target="Test" xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <Success/>
</HTNG_HotelCheckInNotifRS>

```

3.6 Synchronize Cache – Push

3.6.1 Overview

This use case describes the business and system processes and requirements for pushing the data required to perform a full synchronization of in-house reservations, house accounts, membership accounts, etc. and their associated profiles.

Implementation Note:

Any new implementation of synchronization should use the [push protocol detailed below](#) where the cache log is synchronized and “catches up” when the link between the PMS and E-Point of Sale applications is established. This is, however, subject to confirmation by each business user and should be used to complement and enhance rather than replace any current business process.

3.6.2 Roles

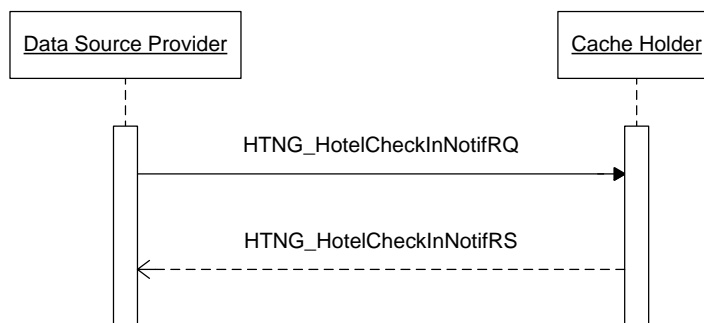
Role	Description	Examples
Cache Holder	A system that caches a view of (current, arriving, or departing) hotel guests and accounts for the purpose of being able to perform local lookups in case communications between systems becomes unavailable.	<ul style="list-style-type: none"> Retail Point of Sale (POS) System Restaurant POS System F&B POS System Spa POS System
Data Source Provider	A system that publishes changes to its data to another system.	Property Management System

3.6.3 Use Case

Assumption:	The Data Source Provider and Cache Holder have established a set of data values that will be exchanged.
Pre-condition:	None

Trigger:	The Data Source Provider has been instructed, either through messaging or manual intervention, that synchronization must take place.
Basic Course of Events:	<ol style="list-style-type: none"> 1) The Data Source Provider iterates through all of the in-house accounts and their associated profiles. 2) The Data Source Provider constructs the appropriate message and transmits it to the Cache Holder.
Post-condition:	The Cache Holder receives the message, parses its content and updates its local cache with the appropriate values.
Exception Path:	The Data Source Provider is unable to push an update to the Cache Holder and must queue the message for later delivery.
Alternative Path:	None

3.6.4 Message Flows



3.6.5 Sample Request

```

<HTNG_HotelCheckInNotifRQ EchoToken="33f07f05-84e5-48ff-b772-894119d5c194" TimeStamp="2011-08-24T09:30:47Z" Version="2.0" Target="Production" xmlns="http://htng.org/2014B"
xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <PropertyInfo ChainCode="STARWOOD" HotelName="Westin San Diego" BrandCode="WESTIN"
HotelCode="2134" HotelCodeContext="CRES"/>
  <AffectedGuests>
    <UniqueID Type="1" ID="GST123"/>
    <UniqueID Type="14" ID="RES123"/>
  </AffectedGuests>
  <Room RoomID="STE100">
    <RoomType RoomID="STE100" InvBlockCode="GRP123"/>
    <TelephoneExtensions>
      <TelephoneExtension>71200</TelephoneExtension>
    </TelephoneExtensions>
  </Room>
  <HotelReservations>
    <ota:HotelReservation ResStatus="Checked In" RoomStayReservation="true">
      <ota:UniqueID Type="14" ID="RES123"/>
      <ota:RoomStays>
        <ota:RoomStay MarketCode="T" DiscountCode="DSC123" RoomStayStatus="In-house">
          <RoomRates>
            <RoomRate RoomTypeCode="KING" RatePlanCode="PKG123" RoomID="101">
              <GuestCounts>
                <GuestCount AgeQualifyingCode="10" Count="2"/>
                <GuestCount AgeQualifyingCode="8" Count="2"/>
              </GuestCounts>
            </RoomRate>
          </RoomRates>
          <ota:TimeSpan End="2011-09-18" Start="2011-09-20"/>
        </ota:RoomStay>
      </ota:RoomStays>
    </ota:HotelReservation>
  </HotelReservations>
</HTNG_HotelCheckInNotifRQ>
  
```

```

</ota: RoomStay>
</ota: RoomStays>
<ota: ResGuests>
  <ResGuest VIP="true" PrimaryIndicator="true">
    <Profiles>
      <ProfileInfo>
        <UniqueID Type="" ID="" ID_Context="" />
        <Profile>
          <Customer>
            <PersonName>
              <NamePrefix>Mr. </NamePrefix>
              <GivenName>Jay</GivenName>
              <Surname>Rosamili</Surname>
            </PersonName>
            <CustLoyalty MembershipID="798654231" ProgramID="HTNG_REWARDS"
Loyal Level="Gold" />
          </Customer>
        </Profile>
      </ProfileInfo>
    </Profiles>
    <Comments>
      <Comment Name="" GuestViable="false">
        <Text Language="en-us" />
      </Comment>
    </Comments>
  </ResGuest>
</ota: ResGuests>
<TPA_Extensions>
  <TPA_Extension>
    <AvailableCredit CurrencyCode="USD" Amount="128.37" DecimalPlaces="0" />
  </TPA_Extension>
</TPA_Extensions>
</ota: HotelReservation>
</HotelReservations>
</HTNG_HotelCheckInNotifRQ>

```

3.6.6 Sample Response

```

<HTNG_HotelCheckInNotifRS EchoToken="a" Timestamp="2001-12-17T09:30:47Z" Version="0.0"
Target="Test" xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <Success/>
</HTNG_HotelCheckInNotifRS>

```

3.7 Synchronize Cache – Pull

3.7.1 Overview

This use case describes the business and system processes and requirements for requesting a resynchronization of to in-house reservations, house accounts, membership accounts, etc. and their associated profiles. This message can be used to initially populate or recover a cache and for subscribers of the [real-time message](#); this pull message can be used as a supplement to ensure synchronization between systems. For those systems that are not able to subscribe to the real-time message, the pull message can be used on a periodic basis to refresh its local cache.

Implementation Note:

Decisions will need to be made by the business owner as to how the business and the interface will handle the scoping of the profile and account data to be synchronized between systems:

- Guests / accounts due to arrive
- Guests / accounts checked out
- Guests / accounts plus shoulder dates
- Long-lived accounts (golf membership, for example)

It is assumed that there is a single cache per interface instance and that the Data Source Provider will provide a consistent mechanism to generate timestamps to indicate data freshness. This value must be used by the Cache Holder to determine if the most current data should be written to its cache.

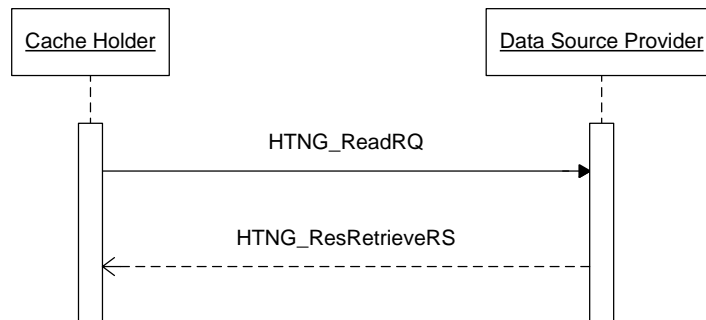
3.7.2 Roles

Role	Description	Examples
Cache Holder	A system that caches a view of (current, arriving, or departing) hotel guests and accounts for the purpose of being able to perform local lookups in case communications between systems becomes unavailable.	<ul style="list-style-type: none">• Retail Point of Sale (POS) System• Restaurant POS System• F&B POS System• Spa POS System
Data Source Provider	A system that publishes changes to its data to another system.	Property Management System

3.7.3 Use Case

Assumption:	The Data Source Provider and Cache Holder have established a set of data values that will be exchanged.
Pre-condition:	The Cache Holder does not have an existing populated cache of data or needs to update the data to the current state as represented by the Data Source Provider.
Trigger:	The Cache Holder has been instructed, either through an event or manual intervention, that synchronization must take place.
Basic Course of Events:	<ol style="list-style-type: none">1) The Cache Holder requests data synchronization from the Data Source Provider.2) The Data Source Provider iterates through the appropriate set of data (as established by trading partners) along with the associated profiles, constructs the appropriate message and transmits it to the Cache Holder.
Post-condition:	The Cache Holder receives the message, parses its content and updates its local cache with the appropriate values.
Exception Path:	None
Alternative Path:	None

3.7.4 Message Flows



3.7.5 Sample Request

```
<HTNG_ReadRQ EchoToken="a" Timestamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
xml ns="http://htng.org/2014B" xml ns:ota="http://www.opentravel.org/OTA/2003/05">
  <POS>
    <ota:Source>
      <ota:RequestorID Type="0" ID_Context="a" ID="a"/>
    </ota:Source>
  </POS>
  <UniqueID Type="14" ID="RES123"/>
  <ReadRequests>
    <HotelReadRequest HotelCode="a" ChainCode="a" BrandCode="a">
      <Verification>
        <ota:ReservationTimeSpan End="2012-01-02" Start="2012-01-03"/>
      </Verification>
    </HotelReadRequest>
  </ReadRequests>
</HTNG_ReadRQ>
```

3.7.6 Sample Response

```
<OTA_ResRetrieveRS EchoToken="a" Timestamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
xml ns="http://www.opentravel.org/OTA/2003/05">
  <Success/>
  <Warnings>
    <Warning Type="0" Status="a" RecordID="a" ShortText="a" Code="0">String</Warning>
  </Warnings>
  <ReservationsList>
    <HotelReservation RoomStayReservation="true" ResStatus="Reserved|In-house|Checked
out|Cancelled">
      <ota:UniqueID Type="14" ID="RES123"/>
      <RoomStays>
        <RoomStay MarketCode="" DiscountCode="" RoomStayStatus="">
          <RatePlans>
            <RatePlan RatePlanCode="PKG123">
              <MealsIncluded MealPlanCodes="MAP"/>
            </RatePlan>
          </RatePlans>
          <RoomRates>
            <RoomRate RoomTypeCode="KING" RoomID="101" InventoryCode="GRP123">
              <GuestCounts>
                <GuestCount AgeQualifyingCode="10" Count="2"/>
                <GuestCount AgeQualifyingCode="8" Count="2"/>
              </GuestCounts>
            </RoomRate>
          </RoomRates>
        </RoomStay>
        <TimeSpan Start="2012-08-20" End="2012-08-21"/>
      </RoomStays>
      <ResGuests>
        <ResGuest VIP="true" PrimaryIndicator="true">
          <Profiles>
            <ProfileInfo>
              <UniqueID Type="" ID="" ID_Context=""/>
            </ProfileInfo>
            <Profile>
              <Customer>
```

```

        <PersonName>
          <NamePrefix>Mr. </NamePrefix>
          <GivenName>Jay</GivenName>
          <Surname>Rosamili</Surname>
        </PersonName>
        <CustLoyalty MembershipID="798654231" ProgramID="HTNG_REWARDS"
Loyal Level="Gold"/>
      </Customer>
    </Profile>
  </ProfileInfo>
</Profiles>
<Comments>
  <Comment Name="" GuestVisible="false">
    <Text Language="en-us"/>
  </Comment>
</Comments>
</ResGuest>
</ResGuests>
<TPA_Extensions>
  <TPA_Extension>
    <AvailableCredit CurrencyCode="USD" Amount="128.37" DecimalPlaces="0"/>
  </TPA_Extension>
</TPA_Extensions>
</HotelReservation>
</ReservationsList>
</OTA_ResRetrievers>

```

3.8 Offline Status Notification

3.8.1 Overview

This use case describes the business and system processes and requirements for establishing that two systems are intentionally placed in an offline mode (for example Point of Sale must go offline for Nightly reports or PMS must go offline during audit or either system will be offline for software maintenance, etc.).

3.8.2 Roles

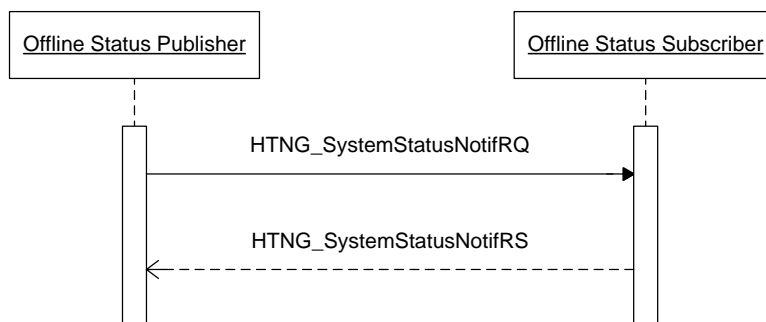
Role	Description	Examples
Offline Status Publisher	A system that notifies another system that it is about to become unavailable and that offline or manual procedures should be taken until system communications resumes.	Property Management System
Offline Status Subscriber	A system that receives a notification from another system that communications is about to become unavailable.	<ul style="list-style-type: none"> Point of Sale System In-room entertainment system PBX

3.8.3 Use Case

Assumptions:	<ul style="list-style-type: none"> The Offline Status Subscriber desires to be notified when the Offline Status Publisher is going offline and communication will be unavailable until further notice. If an offline cache has been implemented, it is current.
--------------	---

Pre-condition:	None
Trigger:	The Offline Status Publisher has encountered a condition which requires it to place itself offline and thus can no longer process or send messages to the other system.
Basic Course of Events:	<ol style="list-style-type: none"> 1) The Offline Status Publisher has determined that it must go offline for some reason. 2) The Offline Status Publisher sends the offline message to the Offline Status Subscriber indicating it is going offline and may include a reason for doing so (examples might be maintenance, system reboot, audit or end-of-day). 3) The Offline Status Subscriber may then take the necessary actions depending on the reason for the offline message or other procedures as configured by the receiving system.
Post-conditions:	<ul style="list-style-type: none"> • The two systems are no longer in communication. • Each system can prepare itself for reconnection as determined by operational parameters.
Exception Path:	None
Alternative Path:	None

3.8.4 Message Flows



3.8.5 Sample Request

```

<HTNG_SystemStatusNotifRQ EchoToken="a" TimeStamp="2013-01-18T23:12:47-05:00" Version="0.0"
Target="Production" xmlns="http://htng.org/2014B"
xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <SystemStatus Status="Offline" ScheduledReturn="2013-01-19T05:12:47-05:00"/>
</HTNG_SystemStatusNotifRQ>
  
```

3.8.6 Sample Response

```

<HTNG_SystemStatusNotifRS EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0"
Target="Test" xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <Success/>
</HTNG_SystemStatusNotifRS>
  
```

3.9 Financial Reconciliation

3.9.1 Overview

This use case describes the business and system processes and requirements for the reconciliation of financial information between systems, typically performed just before or as part of the daily or nightly financial audit process. Please note that one message must be sent per Revenue Center and per interface instance. Please see [Summary Transaction Posting](#) implementation note for more information.

3.9.2 Roles

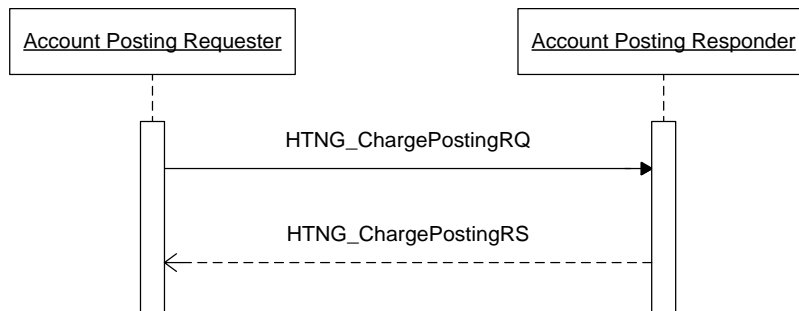
Role Name	Definition	Example
Account Posting Requester	A system that has the need to post charges to guest rooms or accounts.	Charge Posting System
Account Posting Responder	A system that houses guest and account information; receives charge requests and posts them to the appropriate folios.	<ul style="list-style-type: none">Folio/Membership Accounting SystemPMS

3.9.3 Use Case

Assumptions:	<ul style="list-style-type: none">All transaction/charge posting codes and tender types shared between systems have been configured in both the Account Posting Requester and Account Posting Responder.Financial reconciliation has not been run.All transactions must have been sent by the Account Posting Requester and processed by the Account Posting Responder system.
Pre-conditions:	<ul style="list-style-type: none">Any off-line transactions not previously sent during normal daily processing should be sent to the Account Posting Responder prior to initiating this use case.The credits balance with the debits in each message.
Triggers:	<ul style="list-style-type: none">The Account Posting Responder has notified the Account Posting Requester that it has begun the Night Audit Process.Alternatively, the Account Posting Requester manually initiates this use case based on the start of its own Night Audit Process.
Basic Course of Events:	<ol style="list-style-type: none">The Account Posting Requester generates a summary of financial information for a given business day/period, generates the appropriate message payload, and sends it to the Account Posting Responder.The Account Posting Responder verifies that all previously posted room charges and other payment methods are

	accounted for and that the totals match against the specific interface cashier number (set up on installation).
Post-condition:	None
Exception Path:	The data inside the request does not balance with itself.
Alternative Path:	None

3.9.4 Message Flows



3.9.5 Sample Request

```

<HTNG_ChargePostingRQ EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <POS>
    <ota:Source>
      <ota:RequestorID Type="0" ID_Context="a" ID="a">
        <ota:CompanyName CodeContext="a" CompanyShortName="a" Department="a" Division="a"
Code="a"/>
      </ota:RequestorID>
    </ota:Source>
  </POS>
  <PropertyInfo ChainCode="a" HotelName="a" BrandCode="a" HotelCode="a" BrandName="a"
HotelCodeContext="a"/>
  <Posting ID="67436984" BusinessPeriod="2012-02-18">
    <RevenueCenter ID="3" Description="">
      <Terminal ID=""/>
    </RevenueCenter>
    <TotalPostingAmount CurrencyCode="USD" Amount="2512.47" DecimalPlaces="0"/>
    <ReconcilingCheckIDs>
      <ReconcilingCheckID>1001</ReconcilingCheckID>
      <ReconcilingCheckID>1003</ReconcilingCheckID>
      <ReconcilingCheckID>1004</ReconcilingCheckID>
    </ReconcilingCheckIDs>
    <Transaction TicketID="4651" Table="23" Covers="2" OpenTime="2012-12-31T06:23"
CloseTime="2012-12-31T06:52" MealPeriodID="">
      <RevenueDetails>
        <RevenueDetail PMSRevenueCode="" CurrencyCode="USD" Amount="1980.31"
Description="Restaurant Food" DecimalPlaces="0" RevenueCategoryCode="" SubTypeID="">
          <FolioIDs>
            <FolioID>0</FolioID>
          </FolioIDs>
        </RevenueDetail>
        <RevenueDetail PMSRevenueCode="" CurrencyCode="USD" Amount="482.50"
Description="Restaurant Liquor and Wine" DecimalPlaces="0" RevenueCategoryCode="" SubTypeID="">
          <FolioIDs>
            <FolioID>0</FolioID>
          </FolioIDs>
        </RevenueDetail>
        <RevenueDetail ReferenceID="" CurrencyCode="USD" Amount="49.66" Description="Restaurant
Beer" DecimalPlaces="0" RevenueCategoryCode="" SubTypeID="">
          <FolioIDs>
            <FolioID>0</FolioID>
          </FolioIDs>
        </RevenueDetail>
      </RevenueDetails>
    </Transaction>
  </Posting>
</HTNG_ChargePostingRQ>
  
```

```

</RevenueDetails>
<ota:TaxItems CurrencyCode="USD" Amount="304.84" DecimalPlaces="0">
  <ota:Tax CurrencyCode="USD" Amount="183.25" DecimalPlaces="0" Code="0">
    <ota:TaxDescription Name="Retail/Food"/>
  </ota:Tax>
  <ota:Tax CurrencyCode="USD" Amount="117.00" DecimalPlaces="0" Code="0">
    <ota:TaxDescription Name="Liquor/Wine"/>
  </ota:Tax>
</ota:TaxItems>
<Tenders>
  <RevenueDetail ReferenceID="" CurrencyCode="USD" Amount="66.00" Description="Room Charge"
    DecimalPlaces="0" RevenueCategoryCode="6" SubTypeID="RMCHG" TenderID="19">
    <FolioIDs>
      <FolioID>0</FolioID>
    </FolioIDs>
    <Account Type="Account" ID="123456"/>
  </RevenueDetail>
</Tenders>
</Transaction>
</Posting>
</HTNG_ChargePostingRQ>

```

3.9.6 Sample Response

```

<HTNG_ChargePostingRS EchoToken="a" Timestamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
  xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <Success/>
  <PostingGUID>String</PostingGUID>
  <TotalPostingAmount CurrencyCode="USD" Amount="2512.47" DecimalPlaces="0"/>
</HTNG_ChargePostingRS>

```

3.10 Rapid Loyalty Redemption

3.10.1 Overview

Rapid Loyalty Redemption provides program members with a mechanism to use their point balances in lieu of or in conjunction with cash payments for goods and services.

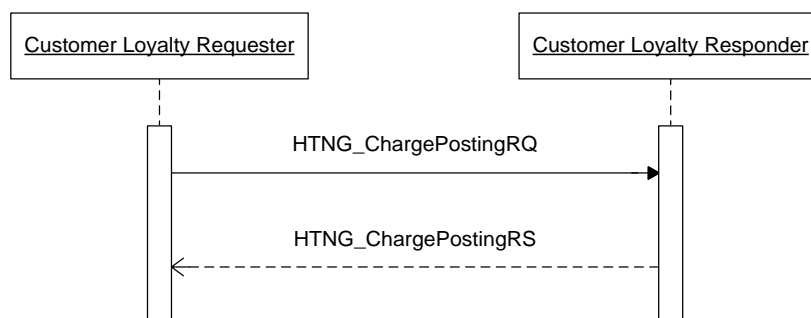
3.10.2 Roles

Role	Description	Example
Customer Loyalty Requester	A system that needs to settle against a loyalty point account.	<ul style="list-style-type: none"> Property Management System Point of Sale (F&B or Retail) Central Reservations System Customer Relationship Management Other Marketing Solutions
Customer Loyalty Responder	A system that manages customer loyalty program memberships and validates transactions.	Loyalty Solution

3.10.3 Use Case

Assumptions:	<ul style="list-style-type: none"> Customer Loyalty Responder maintains point balances for loyalty members. Customer Loyalty Requester may support split-tender. A conversion rate between loyalty points and local currency has been established. Point of Sale system passes the currency or points value.
Pre-conditions:	Customer is a loyalty program member.
Trigger:	Customer Loyalty Requester requests redemption.
Basic Course of Events:	<ol style="list-style-type: none"> 1) Customer Loyalty Requester captures redemption request. 2) Customer Loyalty Requester sends authorization request. 3) Customer Loyalty Responder performs credit check of loyalty member. 4) Customer Loyalty Responder processes transaction. 5) Customer Loyalty Responder issues authorization code for redemption to debit account. 6) Customer Loyalty Requester completes transaction.
Post-conditions:	None
Exception Paths:	<ul style="list-style-type: none"> Member is invalid. Member balance is insufficient.
Alternative Path:	None

3.10.4 Message Flows



3.10.5 Sample Request

```

<HTNG_ChargePostingRQ EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <PostingID="67436984" ZoomInKey="" Void="false" BusinessPeriod="2012-02-18"
Type="LoyaltyAuthorization">
    <CustLoyaltyMembershipID="798654231" ProgramID="HTNG_REWARDS" LoyaltyLevel="Gold"/>
  </PostingID>
</HTNG_ChargePostingRQ>
  
```

```

<Transaction TicketID="4651" Table="23" Covers="2" OpenTime="2012-12-31T06:23"
CloseTime="2012-12-31T06:52" MealPeriodID="">
  <RevenueDetails>
    <RevenueDetail ReferenceID="" CurrencyCode="USD" Amount="8.00" Description="Food"
DecimalPlaces="0" RevenueCategoryCode="14" SubTypeID="Local">
      <FolioIDs>
        <FolioID>0</FolioID>
      </FolioIDs>
    </RevenueDetail>
  </RevenueDetails>
  <ota:TaxItems CurrencyCode="USD" Amount="0.88" DecimalPlaces="0">
    <ota:Tax CurrencyCode="USD" Amount="0.88" DecimalPlaces="0" Code="0">
      <ota:TaxDescription Name="State Tax"/>
    </ota:Tax>
  </ota:TaxItems>
</Transaction>
</Posting>
</HTNG_ChargePostingRQ>

```

3.10.6 Sample Response

```

<HTNG_ChargePostingRS EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <Success/>
  <PostingGUID>String</PostingGUID>
  <AuthorizationResult AuthorizationCode="324532454" ApprovalDateTime="2001-12-17T09:30:48Z"
Result="Approved"/>
  <AccountAuthorization Amount="888" NonISO_CurrencyCode="points"/>
</AuthorizationResult>
  <Accounts>
    <LoyaltyAccountInfo PointBalance="18234"/>
  </Accounts>
</HTNG_ChargePostingRS>

```

3.10.7 Sample Request

```

<HTNG_ChargePostingRQ EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  ...
  <PostingID="67436984" ZoomInKey="" Void="false" BusinessPeriod="2012-02-18"
Type="LoyaltyRedemption">
    <CustLoyalty MembershipID="798654231" ProgramID="HTNG_REWARDS" LoyaltyLevel="Gold"/>
    ...
    <Transaction TicketID="4651" Table="23" Covers="2" OpenTime="2012-12-31T06:23"
CloseTime="2012-12-31T06:52" MealPeriodID="2">
      ...
      <RevenueDetails>
        <RevenueDetail ReferenceID="" CurrencyCode="USD" Amount="8.00" Description="Food"
DecimalPlaces="0" RevenueCategoryCode="14">
          <FolioIDs>
            <FolioID>0</FolioID>
          </FolioIDs>
        </RevenueDetail>
      </RevenueDetails>
      <ota:TaxItems CurrencyCode="USD" Amount="0.88" DecimalPlaces="0">
        <ota:Tax CurrencyCode="USD" Amount="0.88" DecimalPlaces="0" Code="0">
          <ota:TaxDescription Name="State Tax"/>
        </ota:Tax>
      </ota:TaxItems>
      <Tenders>
        <RevenueDetail ReferenceID="324532454" Description="Points Redemption"
RevenueCategoryCode="6" SubTypeID="REDEEM" TenderID="26">
          <FolioIDs>
            <FolioID>0</FolioID>
          </FolioIDs>
          <OtherPaymentForm>
            <LoyaltyRedemption RedemptionQuantity="888"/>
          </OtherPaymentForm>
        </RevenueDetail>
      </Tenders>
    </Transaction>
  </Posting>
</HTNG_ChargePostingRQ>

```

3.10.8 Sample Response

```
<HTNG_ChargePostingRS EchoToken="a" Timestamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <Success/>
  <PostingGUID>String</PostingGUID>
  <TotalPostingAmount CurrencyCode="USD" Amount="8.88" DecimalPlaces="0"/>
  <Accounts>
    <LoyaltyAccountInfo PointBalance="17346"/>
  </Accounts>
</HTNG_ChargePostingRS>
```

3.11 Loyalty Voucher Redemption

3.11.1 Overview

Loyalty Voucher Redemption provides loyalty program members with a mechanism to redeem awarded vouchers for goods or services. A voucher is a reward provided to a loyalty program customer, which may be electronic or paper.

3.11.2 Roles

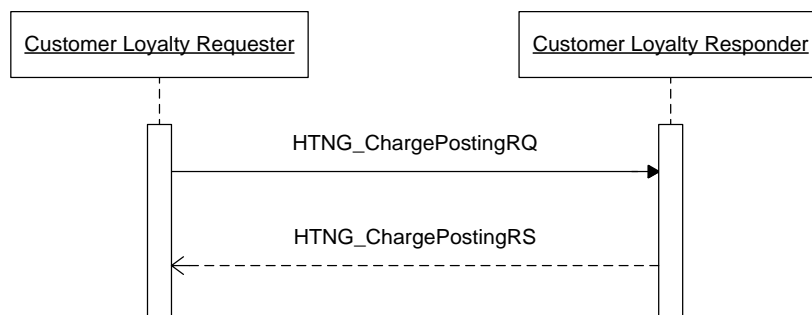
Role	Description	Example
Customer Loyalty Requester	A system that needs to be able to request a new customer loyalty program member enrollment be created.	<ul style="list-style-type: none">Property Management SystemPoint of Sale (F&B or Retail)Central Reservations SystemCustomer Relationship ManagementOther Marketing Solutions
Customer Loyalty Responder	A system that manages customer loyalty program memberships.	Loyalty Systems, such as fast food loyalty programs, hotel loyalty systems, airline frequent flyer programs, etc.

3.11.3 Use Case

Assumptions:	<ul style="list-style-type: none">Customer Loyalty Responder supports redemption of vouchers per account.Customer Loyalty Requester has a way of processing the voucher.
Pre-conditions:	<ul style="list-style-type: none">Customer is a loyalty program member.A voucher (or "reward") has been issued by the Customer Loyalty Responder and is associated with the customer's account.
Trigger:	Customer Loyalty Requestor requests redemption.

Basic Course of Events:	<ol style="list-style-type: none"> 1) Customer is notified of available vouchers that are available to be redeemed. 2) Customer Loyalty Requestor captures redemption request. 3) Customer Loyalty Responder performs credit check of loyalty member. 4) Customer Loyalty Responder approves/confirms redemption. 5) Customer Loyalty Responder issues voucher for redemption to debit account. 6) Customer Loyalty Requestor processes settlement.
Post-conditions:	<ul style="list-style-type: none"> • Member has account balance debited and/or voucher status/availability modified. • Purchase is completed. • Member statement is updated.
Exception Paths:	<ul style="list-style-type: none"> • Member is invalid. • Member available voucher is not valid. • Member balance is insufficient.
Alternative Path:	None

3.11.4 Message Flows



3.11.5 Sample Request

```

<HTNG_ChargePostingRQ EchoToken="abce176" TimeStamp="2001-12-17T09:30:47Z" Version="0.0"
Target="Test" xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <PostingID="67436984" ZoomInKey="" Void="false" BusinessPeriod="2012-02-18">
    <CustLoyaltyMembershipID="798654231" ProgramID="HTNG_REWARDS" LoyalLevel="Gold"/>
    <TransactionTicketID="4651" Table="23" Covers="2" OpenTime="2012-12-31T06:23"
CloseTime="2012-12-31T06:52" MealPeriodID="" DigitsDialled="" Duration="" Extension="">
      <Tenders>
        <RevenueDetail ReferenceID="" CurrencyCode="USD" Amount="66.00" Description="Room Charge"
DecimalPlaces="0" RevenueCategoryCode="6" SubTypeID="RMCHG" TenderID="39">
          <FolioIDs>
            <FolioID>0</FolioID>
          </FolioIDs>
          <OtherPaymentForm>
            <VoucherSeriesCode="11" SupplierIdentifier="CRM123" Identifier="V574142424"/>
          </OtherPaymentForm>
        </RevenueDetail>
      </Tenders>
    </Transaction>
  </PostingID>
</HTNG_ChargePostingRQ>
  
```

```
</RevenueDetail>
</Tenders>
</Transaction>
</Posting>
</HTNG_ChargePostingRQ>
```

3.11.6 Sample Response

```
<HTNG_ChargePostingRS EchoToken="abce176" TimeStamp="2001-12-17T09:30:47Z" Version="0.0"
Target="Test" xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <Success/>
  <PostingGUID>String</PostingGUID>
  <TotalPostingAmount CurrencyCode="USD" Amount="66.00" DecimalPlaces="0"/>
  <Vouchers>
    <Voucher EffectiveDate="2012-01-01" ExpiryDate="2012-12-31" SeriesCode="11"
SupplierIdentifier="CRM123" Identifier="V574142424" Amount="34.00" CurrencyCode="USD"/>
  </Vouchers>
</HTNG_ChargePostingRS>
```

3.12 Loyalty Balance Inquiry

3.12.1 Overview

This use case describes processes for a user to check a Loyalty Member's account balance from a Point of Sale workstation. This balance may be represented as currency, points, vouchers or offers.

3.12.2 Roles

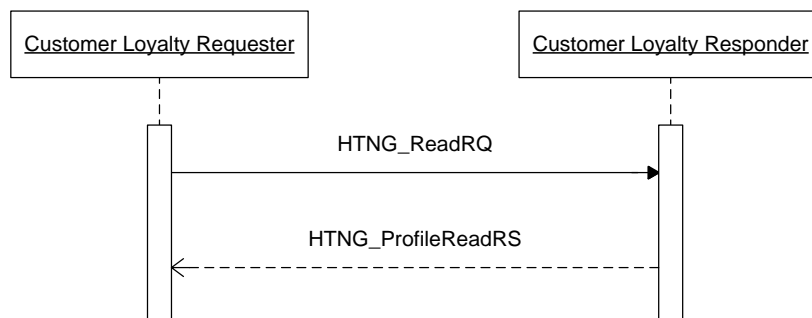
Role	Description	Example
Customer Loyalty Requester	A system that needs to be able to request a new customer loyalty program member enrollment be created.	<ul style="list-style-type: none"> Property Management System Point of Sale (F&B or Retail) Central Reservations System Customer Relationship Management Other Marketing Solutions
Customer Loyalty Responder	A system that manages customer loyalty program memberships.	Loyalty Solution

3.12.3 Use Case

Assumptions:	<ul style="list-style-type: none"> Customer Loyalty Responder maintains a balance for loyalty members. A look up button exists on the workstation that is accessible by the user. POS has the ability to display to the user values including currency, points, voucher or offer details.
Pre-conditions:	Customer is a loyalty program member.

Trigger:	The Customer Loyalty Requester has the need to identify the balance on an account.
Basic Course of Events:	<ol style="list-style-type: none"> 1) Customer Loyalty Requestor captures membership identification through magnetic swipe, RFID, bar code, manually entry or other capture method. 2) Customer Loyalty Requestor sends a request to the Customer Loyalty Responder containing membership identification. 3) Customer Loyalty Responder searches the database and locates matching accounts, their associated balances and level/status of membership (Gold, Silver, VIP). 4) Customer Loyalty Responder sends a message containing the matching balance(s)/offer(s) and level/status to the Customer Loyalty Requestor.
Post-conditions:	Customer Loyalty Requester may elect to display or print the available currency, points, or offers and level/status.
Exception Paths:	Member is invalid.
Alternative Path:	None

3.12.4 Message Flows



3.12.5 Sample Request

```

<HTNG_ReadRQ EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
xml ns="http://htng.org/2014B" xml ns:ota="http://www.opentravel.org/OTA/2003/05">
  <RequestedComponents>
    <RequestedComponent Name="CustomerName"/>
    <RequestedComponent Name="CustomerVouchers"/>
    <RequestedComponent Name="Extension" Extension="TPA_Extensions"/>
  </RequestedComponents>
  <POS>
    <ota:Source>
      <ota:RequestorID Type="0" ID_Context="a" ID="a"/>
    </ota:Source>
  </POS>
  <UniqueID Type="0" ID_Context="a" ID="a"/>
  <ReadRequests>
    <HotelReadRequest HotelCode="a" ChainCode="a" BrandCode="a">
      <Verification>

```

```

    <ota: PersonName PartialName="true">
      <ota: GivenName>a</ota: GivenName>
      <ota: Surname>a</ota: Surname>
    </ota: PersonName>
    <ota: CustLoyalty MembershipID="798654231" ProgramID="HTNG_REWARDS" Loyal Level="Gold"/>
    <ota: ReservationTimeSpan End="1967-08-13" Start="1967-08-13"/>
    <Room RoomID="101"/>
  </Verification>
</HotelReadRequest>
</ReadRequests>
</HTNG_ReadRQ>

```

3.12.6 Sample Response

```

<OTA_ResRetrieveRS EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
xmlns="http://www.opentravel.org/OTA/2003/05">
  <Success/>
  <Warnings>
    <Warning Type="0" Status="a" RecordID="a" ShortText="a" Code="0">String</Warning>
  </Warnings>
  <ReservationsList>
    <HotelReservation RoomStayReservation="true" ResStatus="Reserved|In-house|Checked
out|Cancelled">
      <RoomStays>
        <RoomStay MarketCode="T" DiscountCode="DSC123" RoomStayStatus="In-house">
          <RoomRates>
            <RoomRate RoomTypeCode="KING" RatePlanCode="PKG123" RoomID="101">
              <GuestCounts>
                <GuestCount AgeQualifyingCode="10" Count="2"/>
                <GuestCount AgeQualifyingCode="8" Count="2"/>
              </GuestCounts>
            </RoomRate>
          </RoomRates>
        </RoomStay>
        <TimeSpan Start="2012-08-20" End="2012-08-21"/>
      </RoomStays>
      <ResGuests>
        <ResGuest VIP="true" PrimaryIndicator="true">
          <Profiles>
            <ProfileInfo>
              <UniqueID Type="" ID="" ID_Context=""/>
            <Profile>
              <Customer>
                <PersonName>
                  <NamePrefix>Mr. </NamePrefix>
                  <GivenName>Jay</GivenName>
                  <Surname>Rosamili</Surname>
                </PersonName>
                <PaymentForm>
                  <Voucher EffectiveDate="2012-01-01" ExpireDate="2012-12-31" SeriesCode="11"
SupplierIdentifier="CRM123" Identifier="V574142424" IssueReason="Loyalty" DeliveryMethod="Email"
MonetaryValue="100.00" CurrencyCode="USD" Status="Valid" Remark=""/>
                </PaymentForm>
                <CustLoyalty MembershipID="798654231" ProgramID="HTNG_REWARDS"
Loyal Level="Gold">
                  <SubAccountBalance Type="points" Balance="56151"/>
                </CustLoyalty>
                <TPA_Extensions>
                  <TPA_Extension>
                    <LoyaltyTierProgress>
                      <Accrual CurrentTier="GOLD" Effective="2012-01-01" Expire="2012-12-31">
                        <Current Accrual Type="Points" Value="1000"/>
                        <Current Accrual Type="Nights" Value="2"/>
                        <Lifetime Accrual Type="Points" Value="169000"/>
                        <Lifetime Accrual Type="Nights" Value="34"/>
                      </Accrual>
                    <NextTiers>
                      <NextTier Tier="SILVER">
                        <Milestone Accrual Type="Points" Value="5000" Needed="4000"/>
                        <Milestone Accrual Type="Nights" Value="5" Needed="3"/>
                      </NextTier>
                      <NextTier Tier="GOLD">
                        <Milestone Accrual Type="Points" Value="10000" Needed="9000"/>
                        <Milestone Accrual Type="Nights" Value="10" Needed="8"/>
                      </NextTier>
                      <NextTier Tier="PLATINUM">
                        <Milestone Accrual Type="Points" Value="20000" Needed="19000"/>
                        <Milestone Accrual Type="Nights" Value="20" Needed="18"/>
                      </NextTier>
                    </NextTiers>
                  </TPA_Extension>
                </LoyaltyTierProgress>
              </Profile>
            </ProfileInfo>
          </Profiles>
        </ResGuest>
      </ResGuests>
    </HotelReservation>
  </ReservationsList>
</OTA_ResRetrieveRS>

```

```
</NextTiers>
</LoyaltyTierProgress>
</TPA_Extension>
</TPA_Extensions>
</Customer>
</Profile>
</ProfileInfo>
</Profiles>
<Comments>
  <Comment Name="" GuestViable="false">
    <Text Language="en-us"/>
  </Comment>
</Comments>
</ResGuest>
</ResGuests>
<TPA_Extensions>
  <TPA_Extension>
    <AvailableCredit CurrencyCode="USD" Amount="128.37" DecimalPlaces="0"/>
  </TPA_Extension>
</TPA_Extensions>
</HotelReservation>
</ReservationsList>
</OTA_ResRetrieveRS>
```

3.13 Associating Transactions with Loyalty Number

There are two means by which a system can obtain the transaction detail from the Point of Sale system for the purpose of calculating points for a given loyalty member.

The first is via the HTNG_ChargePostingRQ message found in this Point of Sale specification.

The second is via the HTNG_HotelFolioNotifRQ and HTNG_HotelFolioRQ messages found in the [HTNG Folio Detail Exchange Specification](#).

Implementers would likely want to implement one of these methods; not both.

If the Customer Loyalty System is fed transaction information by all systems, the HTNG_ChargePostingRQ message can be used. Keep in mind that this message would still need to be sent to another system for any non-cash or credit tenders (e.g. Room Charges, Membership, etc.).

If the Customer Loyalty System receives its information from a Property Management System, only the Property Management System should receive the HTNG_ChargePostingRQ message. If any other system were to receive and process this message, loyalty members could be double-credited for points unless sophisticated logic was built into either the Property Management System and/or the Customer Loyalty System. In this case, the Property Management System and the Customer Loyalty System should implement only the HTNG_HotelFolio(Notif)RQ message (and have the POS to send the HTNG_ChargePostingRQ message only to the PMS).

From Folio Detail Exchange Spec:

```
<HTNG_HotelFolioNotifRQ EchoToken="e23a0dab-9a03-4ab6-8c8e-7b9cb68fdf2d" Timestamp="2010-02-12T12:26:47" Version="1.0" xmlns="http://htng.org/2014B"
xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  <UniqueID Type="14" ID="RES123456"/>
  <Folios>
    <Folio>
      <CustomerProfile>
        <ota:Customer>
```

```
    ...
    <CustLoyal ty Membershi pID="798654231" ProgramID="HTNG_REWARDS" Loyal Level ="Gol d" />
  </ota: Customer>
</CustomerProfile>
<PayerProfile>
  <ota: Customer>
    ...
    <CustLoyal ty Membershi pID="798654231" ProgramID="HTNG_REWARDS" Loyal Level ="Gol d" />
  </ota: Customer>
</PayerProfile>
</Fol i o>
</Fol ios>
</HTNG_Hotel Fol i oNoti fRQ>
```

From this spec:

```
<HTNG_ChargePostingRQ EchoToken="a" TimeStamp="2001-12-17T09:30:47Z" Version="0.0" Target="Test"
xmlns="http://htng.org/2014B" xmlns:ota="http://www.opentravel.org/OTA/2003/05">
  ...
  <PropertyInfo ChainCode="a" Hotel Name="a" BrandCode="a" Hotel Code="a" BrandName="a"
Hotel CodeContext="a"/>
  <Posting ID="67436984" ZoomInKey="" Void="false" BusinessPeriod="2012-02-18">
    ...
    <CustLoyal ty Membershi pID="798654231" ProgramID="HTNG_REWARDS" Loyal Level ="Gol d" />
  </Posting>
</HTNG_ChargePostingRQ>
```

4 Messages

4.1 Account Lookup / Synchronize Cache – Pull

4.1.1 Data Element Table – Request

Element @Attribute	Num	Description/Contents
HTNG_ReadRQ	1	Root element of the message.
@EchoToken	1	A reference for additional message identification, assigned by the requesting host system. When a request message includes an echo token, the corresponding response message MUST include an echo token with an identical value.
@TimeStamp	1	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).
@Version	1	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.
HTNG_ReadRQ / POS / Source	1	This holds details regarding the requestor. It may be repeated to also accommodate the delivery systems.
HTNG_ReadRQ / POS / Source / RequestorID	1	An identifier of the entity making the request (e.g., ATA/IATA/ID number, Electronic Reservation Service Provider (ERSP), Association of British Travel Agents (ABTA)).
@Type	0..1	A reference to the type of object defined by the UniqueID element. Refer to OpenTravel Code List Unique ID Type (UIT).
@ID_Context	0..1	Used to identify the source of the identifier (e.g., IATA, ABTA).
@ID	1	A unique identifying value assigned by the creating system. The ID attribute may be used to reference a primary-key value within a database or in a particular implementation.
HTNG_ReadRQ / UniqueID	0..1	An identifier used to uniquely reference an object in a system (e.g., an airline reservation reference, customer profile reference, booking confirmation number, or a reference to a previous availability quote).

Element @Attribute	Num	Description/Contents
@ID	1	A unique identifying value assigned by the creating system. The ID attribute may be used to reference a primary-key value within a database or in a particular implementation.
@Type	0..1	A reference to the type of object defined by the UniqueID element. Refer to OpenTravel Code List Unique ID Type (UIT).
@ID_Context	0..1	Used to identify the source of the identifier (e.g., IATA, ABTA).
HTNG_ReadRQ / ReadRequests	1	A grouping of Read Requests by travel vertical.
HTNG_ReadRQ / ReadRequests / HotelReadRequest	1	To request a profile when the profile identifier is not known.
@HotelCode	0..1	Describes whether the line item refers to an individual Hotel property.
@ChainCode	0..1	Describes whether the line item refers to a Chain (e.g., Courtyard, Doubletree).
@BrandCode	0..1	Describes whether the line item refers to a Brand (e.g., Marriott, Hilton).
HTNG_ReadRQ / ReadRequests / HotelReadRequest / UniqueID	0..1	An identifier used to uniquely reference an object in a system (e.g., an airline reservation reference, customer profile reference, booking confirmation number, or a reference to a previous availability quote).
@ID	1	A unique identifying value assigned by the creating system. The ID attribute may be used to reference a primary-key value within a database or in a particular implementation.
@Type	1	A reference to the type of object defined by the UniqueID element. Refer to OpenTravel Code List Unique ID Type (UIT).
@ID_Context	0..1	Used to identify the source of the identifier (e.g., IATA, ABTA).
HTNG_ReadRQ / ReadRequests / HotelReadRequest / Verification	1	This is to be used for verification that the record to be returned is the specific record requested.
HTNG_ReadRQ / ReadRequests / HotelReadRequest / Verification / PersonName	0..1	Detailed name information for the customer.
@PartialName	0..1	When true, the full name is not provided.

Element @Attribute	Num	Description/Contents
HTNG_ReadRQ / ReadRequests / HotelReadRequest / Verification / PersonName / GivenName	0..1	Given name, first name or names.
HTNG_ReadRQ / ReadRequests / HotelReadRequest / Verification / PersonName / Surname	1	Family name, last name. May also be used for full name if the sending system does not have the ability to separate a full name into its parts; e.g. the surname element may be used to pass the full name.
HTNG_ReadRQ / ReadRequests / HotelReadRequest / Verification / CustLoyalty	0..1	Loyalty program information for the customer.
@MembershipID	1	Unique identifier of the member in the program (membership number, account number, etc.).
@VendorCode	0..1	Indicate the partner(s)/vendor(s) for which the customer loyalty number is valid.
@ProgramID	0..1	Identifier to indicate the company owner of the loyalty program.
HTNG_ReadRQ / ReadRequests / HotelReadRequest / Verification / ReservationTimespan	0..1	The start and end date of the reservation.
@Start	1	The starting value of the time span.
@End	1	The ending value of the time span.
HTNG_ReadRQ / ReadRequests / HotelReadRequest / Verification / Room	0..1	Used to convey information about a single room or a suite comprised of room components. The populated values on this entity are used as the query parameters.
@RoomID	1	A string value representing the unique identification of a room.

4.1.2 Data Element Table – Response

Element @Attribute	Num	Description/Contents
OTA_ResRetrieveRS	1	Root element of the message.
@EchoToken	1	A reference for additional message identification, assigned by the requesting host system. When a request message includes an echo token, the corresponding response message MUST include an echo token with an identical value.

Element @Attribute	Num	Description/Contents
@TimeStamp	1	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).
@Version	1	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.
@Target	0..1	Used to indicate whether the request is for the Test or Production system.
OTA_ResRetrieveRS / Success	0..1	The presence of the empty Success element explicitly indicates that the OpenTravel versioned message succeeded.
OTA_ResRetrieveRS / Warnings	0..1	Used in conjunction with the Success element to define one or more business errors.
OTA_ResRetrieveRS / Warnings / Warning	1..n	Used when a message has been successfully processed to report any warnings or business errors that occurred.
@Type	1	The Warning element MUST contain the Type attribute that uses a recommended set of values to indicate the warning type. The validating XSD can expect to accept values that it has NOT been explicitly coded for and process them by using Type = "Unknown". Refer to OpenTravel Code List Error Warning Type (EWT).
@Status	0..1	If present, recommended values are those enumerated in the OTA_ErrorRS, (NotProcessed Incomplete Complete Unknown) however, the data type is designated as string data, recognizing that trading partners may identify additional status conditions not included in the enumeration.
@ShortText	1	An abbreviated version of the error in textual format.
@Code	0..1	If present, this refers to a table of coded values exchanged between applications to identify errors or warnings. Refer to OpenTravel Code List Error Codes (ERR).

Element @Attribute	Num	Description/Contents
OTA_ResRetrieveRS / RoomInformationList	0..1	The result set generated by the query sent in the request.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation		One line in a list of hotel reservations. It may contain the reservation id, booked date, arrival and departure dates, number of nights and rooms, hotel info, guest info, and room info.
@RoomStayReservation	0..1	Boolean is True if this reservation is reserving rooms; False if it is only reserving services.
@ResStatus	1	Indicates the status of the reservation.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / RoomStays	0..1	Collection of room stays.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / RoomStays / RoomStay	1..n	Details on the Room Stay including Guest Counts, Time Span of this Room Stay, pointers to Res Guests, guest Memberships, Comments and Special Requests.
@MarketCode	0..1	The code that relates to the market being sold to (e.g., the corporate market, packages).
@DiscountCode	0..1	A discount code known to the hotel.
@RoomStayStatus	0..1	Identifies the status of the room stay.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / RoomStays / RoomStay / RoomRates	0..1	A collection of Room Rates associated with a particular Room Stay.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / RoomStays / RoomStay / RoomRates / RoomRate	1..n	Individual rate amount. This rate is valid for a range of number of occupants and an occupant type.
@RoomTypeCode	0..1	Specific system room type code, ex: A1K, A1Q, etc.
@RatePlanCode	0..1	A string value may be used to request a particular code or an ID if the guest qualifies for a specific rate, such as AARP, AAA, a corporate rate, etc., or to specify a negotiated code as a result of a negotiated rate.

Element @Attribute	Num	Description/Contents
@RoomID	0..1	A string value representing the unique identification of a room.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / RoomStays / RoomStay / RoomRates / RoomRate / GuestCounts	0..1	A collection of Guest Counts associated with the room rate.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / RoomStays / RoomStay / RoomRates / RoomRate / GuestCounts / GuestCount	1..n	A recurring element that identifies the number of guests and ages of the guests.
@AgeQualifyingCode	0..1	A code representing a business rule that determines the charges for a guest based upon age range (e.g., Adult, Child, Senior, Child With Adult, Child Without Adult). This attribute allows for an increase in rate by occupant class. Refer to OpenTravel Code List Age Qualifying Code (AQC).
@Count	0..1	The number of guests in one AgeQualifyingCode or Count.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / RoomStays / RoomStay / Comments	0..1	A collection of Comment objects. Comments which apply to the whole Reservation or a particular Room Stay or Service.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / RoomStays / RoomStay / Comments / Comment	1..n	Comment details.
@Name	0..1	In many cases the description repeats, this will allow you to define the information that is being sent, typically used when.
@GuestViewable	0..1	When true, the comment may be shown to the consumer. When false, the comment may not be shown to the consumer.

Element @Attribute	Num	Description/Contents
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / RoomStays / RoomStay / Comments / Comment / Text	1	Formatted text content.
@Language	0..1	Language identification.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / RoomStays / RoomStay / TimeSpan	0..1	The Time Span which covers the Room Stay.
@Start	1	The starting value of the time span.
@End	1	The ending value of the time span.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / ResGuests	0..1	Collection of guests associated with the reservation.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / ResGuests / ResGuest	1..n	The ResGuest object contains the information about a guest associated with a reservation.
@VIP	0..1	Guest is VIP indicator.
@PrimaryIndicator	0..1	When true indicates this is the primary guest.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / ResGuests / ResGuest / Profiles	1	A collection of Profile objects or Unique IDs of Profiles.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo	1..n	A collection of Profiles or Unique IDs of Profiles.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo / UniqueID	1	A unique ID for a Profile. This element repeats to accommodate multiple unique IDs for a single Profile across multiple systems.

Element @Attribute	Num	Description/Contents
@ID	1	A unique identifying value assigned by the creating system. The ID attribute may be used to reference a primary-key value within a database or in a particular implementation.
@Type	1	A reference to the type of object defined by the UniqueID element. Refer to OpenTravel Code List Unique ID Type (UIT).
@ID_Context	0..1	Used to identify the source of the identifier (e.g., IATA, ABTA).
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo / Profile	0..1	Provides detailed information regarding either a company or a customer Profile.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo / Profile / Customer	0..1	Detailed customer information for this Profile.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo / Profile / Customer / PersonName	1	Detailed name information for the customer.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo / Profile / Customer / PersonName / NamePrefix	0..1	The Salutation for the name. This SHOULD be a value representing an individual (Mr., Mrs., Dr.) and not Mr. and Mrs..
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo / Profile / Customer / PersonName / GivenName	0..1	Given name, first name or names.

Element @Attribute	Num	Description/Contents
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo / Profile / Customer / PersonName / SurName	0..1	Family name, last name. May also be used for full name if the sending system does not have the ability to separate a full name into its parts, e.g. the surname element may be used to pass the full name.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo / Profile / Customer / CustLoyalty	0..1	Loyalty program information for the customer.
@MembershipID	1	Unique identifier of the member in the program (membership number, account number, etc.).
@VendorCode	0..1	Indicate the partner(s)/vendor(s) for which the customer loyalty number is valid.
@ProgramID	0..1	Identifier to indicate the company owner of the loyalty program.
@LoyalLevel	0..1	Indicates special privileges in program assigned to individual.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / TPA_Extensions	0..1	A placeholder in the schema to allow for additional elements and attributes to be included if required, per Trading Partner Agreement (TPA).
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / TPA_Extensions / TPA_Extension	0..n	A single instance of a Trading Partner Agreement extension.
OTA_ResRetrieveRS / RoomInformationList / HotelReservation / TPA_Extensions / TPA_Extension / AvailableCredit	0..1	Used to convey information about the available credit of an account.
@Amount	1	The total amount of available credit as determined by the responding system.
@CurrencyCode	0..1	The code specifying a monetary unit. Use ISO 4217, three alpha code.

Element @Attribute	Num	Description/Contents
@DecimalPlaces	0..1	Indicates the number of decimal places for a particular currency. This is equivalent to the ISO 4217 standard "minor unit". Typically used when the amount provided includes the minor unit of currency without a decimal point (e.g., USD 8500 needs DecimalPlaces="2" to represent \$85).

4.2 Charge / Void Posting / Financial Reconciliation / Rapid Loyalty Redemption / Loyalty Voucher Redemption

4.2.1 Data Element Table – Request

Element @Attribute	Num	Description/Contents
HTNG_ChargePostingRQ	1	Root element of the message.
@EchoToken	1	A reference for additional message identification, assigned by the requesting host system. When a request message includes an echo token, the corresponding response message MUST include an echo token with an identical value.
@TimeStamp	1	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).
@Version	1	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.
HTNG_ChargePostingRQ / POS / Source	1	This holds details regarding the requestor. It may be repeated to also accommodate the delivery systems.
HTNG_ChargePostingRQ / POS / Source / RequestorID	1	An identifier of the entity making the request (e.g., ATA/IATA/ID number, Electronic Reservation Service Provider (ERSP), Association of British Travel Agents (ABTA)).
@Type	0..1	A reference to the type of object defined by the UniqueID element. Refer to OpenTravel Code List Unique ID Type (UIT).
@ID_Context	0..1	Used to identify the source of the identifier (e.g., IATA, ABTA).

Element @Attribute	Num	Description/Contents
@ID	1	A unique identifying value assigned by the creating system. The ID attribute may be used to reference a primary-key value within a database or in a particular implementation.
HTNG_ChargePostingRQ / POS / Source / RequestorID / CompanyName	0..1	Identifies the company that is associated with the UniqueID.
@CompanyShortName	1	Used to provide the company common name.
@Department	0..1	The department name or ID with which the contact is associated.
@Division	0..1	The division name or ID with which the contact is associated.
HTNG_ChargePostingRQ / PropertyInfo	0..1	Information pertaining to a given hotel property.
@HotelCode	1	Describes whether the line item refers to an individual Hotel property.
@ChainCode	0..1	Describes whether the line item refers to a Chain (e.g., Courtyard, Doubletree).
@BrandCode	0..1	Describes whether the line item refers to a Brand (e.g., Marriott, Hilton).
HTNG_ChargePostingRQ / Posting	1	Describes the charge posting and associated information.
@ID	1	A unique identifier for this posting.
@ZoomInKey	0..1	Used by the HTNG_CheckDisplayRQ message to retrieve a preformatted version of the charge posting.
@Void	0..1	Indicates the desire to void a previously posted transaction. This value is assumed to be "false" if not present.
@Type	0..1	For use with the process of redeeming loyalty points. Valid values are <ul style="list-style-type: none"> LoyaltyAuthorization LoyaltyRedemption
HTNG_ChargePostingRQ / Posting / RevenueCenter	1	Information pertaining to the revenue center that generated the posting.
@ID	1	The identifier for the revenue center.

Element @Attribute	Num	Description/Contents
@Description	0..1	The revenue center description.
HTNG_ChargePostingRQ / Posting / TotalPostingAmount	0..1	An amount pertaining to the overall posting.
@Amount	1	The sum of all line items, taxes, gratuities, etc.
@DecimalPlaces	0..1	Indicates the number of decimal places for a particular currency. This is equivalent to the ISO 4217 standard "minor unit." Typically used when the amount provided includes the minor unit of currency without a decimal point (e.g., USD 8500 needs DecimalPlaces="2" to represent \$85).
@CurrencyCode	0..1	The code specifying a monetary unit. Use ISO 4217, three alpha code.
HTNG_ChargePostingRQ / Posting / ServerInfo	0..1	Information about the server.
@EmployeeId	1	The employee ID for the server.
HTNG_ChargePostingRQ / Posting / CashierInfo	0..1	Information about the cashier.
@EmployeeId	1	The employee ID for the cashier.
HTNG_ChargePostingRQ / Posting / CustLoyalty	0..1	Loyalty program information.
@MembershipID	1	Unique identifier of the member in the program (membership number, account number, etc.).
@ProgramID	0..1	Identifier to indicate the company owner of the loyalty program.
@LoyalLevel	0..1	Indicates special privileges in program assigned to individual.
HTNG_ChargePostingRQ / Posting / Transaction	1	Transactional information pertaining to a check/invoice.
@TicketID	1	The check/invoice number representing the charges.
@Table	0..1	The table served.
@Covers	0..1	The number of individuals seated at the table.
@OpenTime	0..1	Open time of the check.
@CloseTime	0..1	Close time of the check.

Element @Attribute	Num	Description/Contents
@MealPeriod	0..1	Meal period.
@DigitsDialed	0..1	The digits dialed to complete a telephone call.
@Duration	0..1	The length of a telephone call.
@Extension	0..1	The extension that made the telephone call.
HTNG_ChargePostingRQ / Posting / Transaction / RevenueDetails	1	A collection of line items.
HTNG_ChargePostingRQ / Posting / Transaction / RevenueDetails / RevenueDetail	1..n	The line item detail of specific revenue transactions.
@ReferenceID	0..1	The unique transaction identifier for this posting.
@Amount	1	A monetary amount.
@DecimalPlaces	0..1	Indicates the number of decimal places for a particular currency. This is equivalent to the ISO 4217 standard "minor unit." Typically used when the amount provided includes the minor unit of currency without a decimal point (e.g., USD 8500 needs DecimalPlaces="2" to represent \$85).
@CurrencyCode	0..1	The code specifying a monetary unit. Use ISO 4217, three alpha code.
@Description	1	The line item detail description for this posting.
@RevenueCategoryCode	1	Describes the type of revenue generated. Refer to OpenTravel Code List Revenue Category Code (RCC).
@SubTypeID	0..1	A value used to further detail the category.
@PMSRevenueCode	0..1	The transaction code assigned by the PMS.
HTNG_ChargePostingRQ / Posting / Transaction / RevenueDetails / RevenueDetail / ExtendedPrice	0..1	The total amount charged for the service including additional amounts and fees.
@AmountBeforeTax	1	The unit amount not including any associated tax (e.g., sales tax, VAT, GST or any associated tax).

Element @Attribute	Num	Description/Contents
@DecimalPlaces	0..1	Indicates the number of decimal places for a particular currency. This is equivalent to the ISO 4217 standard "minor unit." Typically used when the amount provided includes the minor unit of currency without a decimal point (e.g., USD 8500 needs DecimalPlaces="2" to represent \$85).
@CurrencyCode	0..1	The code specifying a monetary unit. Use ISO 4217, three alpha code.
@Quantity	1	The number of items purchased.
HTNG_ChargePostingRQ / Posting / Transaction / TaxItems	1	A collection of taxes.
@Amount	1	A monetary amount.
@DecimalPlaces	0..1	Indicates the number of decimal places for a particular currency. This is equivalent to the ISO 4217 standard "minor unit." Typically used when the amount provided includes the minor unit of currency without a decimal point (e.g., USD 8500 needs DecimalPlaces="2" to represent \$85).
@CurrencyCode	0..1	The code specifying a monetary unit. Use ISO 4217, three alpha code.
HTNG_ChargePostingRQ / Posting / Transaction / TaxItems / Tax	1..n	An individual tax.
@Code	1	Code identifying the fee (e.g., agency fee, municipality fee). Refer to OpenTravel Code List Fee Tax Type (FTT).
@Amount	1	A monetary amount.
@DecimalPlaces	0..1	Indicates the number of decimal places for a particular currency. This is equivalent to the ISO 4217 standard "minor unit." Typically used when the amount provided includes the minor unit of currency without a decimal point (e.g., USD 8500 needs DecimalPlaces="2" to represent \$85).
@CurrencyCode	0..1	The code specifying a monetary unit. Use ISO 4217, three alpha code.

Element @Attribute	Num	Description/Contents
HTNG_ChargePostingRQ / Posting / Transaction / TaxItems / Tax / TaxDescription	0..1	Text description of the taxes in a given language.
@Name	1	In many cases the description repeats, this will allow you to define the information that is being sent, typically used when multiple Comment elements are being sent
HTNG_ChargePostingRQ / Posting / Transaction / Tenders	1	A collection of tenders.
HTNG_ChargePostingRQ / Posting / Transaction / Tenders / RevenueDetail	1..n	The line item detail of specific revenue transactions.
@ReferenceID	0..1	The unique transaction identifier for this posting. If this is a Loyalty Redemption transaction, this is the authorization code provided by the loyalty system.
@Amount	1	A monetary amount.
@DecimalPlaces	0..1	Indicates the number of decimal places for a particular currency. This is equivalent to the ISO 4217 standard "minor unit." Typically used when the amount provided includes the minor unit of currency without a decimal point (e.g., USD 8500 needs DecimalPlaces="2" to represent \$85).
@CurrencyCode	0..1	The code specifying a monetary unit. Use ISO 4217, three alpha code.
@Description	1	The line item detail description for this posting.
@RevenueCategoryCode	1	Describes the type of revenue generated. Refer to OpenTravel Code List Revenue Category Code (RCC).
@SubTypeID	0..1	A value used to further detail the category.
@PMSRevenueCode	0..1	The transaction code assigned by the PMS.
@TenderID	1	A code determined by the sending system used to identify the tender/payment type used to settle a portion or all of the check.
HTNG_ChargePostingRQ / Posting / Transaction / Tenders / RevenueDetail / Account	0..1	The account for which a charge will be applied.

Element @Attribute	Num	Description/Contents
@Type	1	The type of account. Possible values are "Account", "Room" "Extension".
@ID	1	An identifier representing the account for which an amount will be posted.
HTNG_ChargePostingRQ / Posting / Transaction / Tenders / RevenueDetail / Other_PaymentForm	0..1	Used when posting to something other than an Account, Room or telephone extension.
HTNG_ChargePostingRQ / Posting / Transaction / Tenders / RevenueDetail / Other_PaymentForm / LoyaltyRedemption	0..1	Details of a loyalty redemption arrangement. This is normally miles or points.
@RedemptionQuantity	1	The quantity of loyalty units being redeemed.
HTNG_ChargePostingRQ / Posting / Transaction / Tenders / RevenueDetail / Other_PaymentForm / Voucher	0..1	Details of a paper or electronic document indicating prepayment.
@SeriesCode	0..1	Identification of a series of coupons or vouchers identified by serial number(s).
@SupplierIdentifier	0..1	Unique identifier of the electronic voucher, created by the supplier.
@Identifier	1	Unique identifier of the electronic voucher.

4.2.2 Data Element Table – Response

Element @Attribute	Num	Description/Contents
HTNG_ChargePostingRS	1	Root element of the message.
@EchoToken	1	A reference for additional message identification, assigned by the requesting host system. When a request message includes an echo token, the corresponding response message MUST include an echo token with an identical value.

Element @Attribute	Num	Description/Contents
@TimeStamp	1	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).
@Version	1	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.
@Target	0..1	Used to indicate whether the request is for the Test or Production system.
HTNG_ChargePostingRS / Success	0..1	The presence of the empty Success element explicitly indicates that the OpenTravel versioned message succeeded.
HTNG_ChargePostingRS / Warnings	0..1	Used in conjunction with the Success element to define one or more business errors.
HTNG_ChargePostingRS / Warnings / Warning	1..n	Used when a message has been successfully processed to report any warnings or business errors that occurred.
@Type	1	The Warning element MUST contain the Type attribute that uses a recommended set of values to indicate the warning type. The validating XSD can expect to accept values that it has NOT been explicitly coded for and process them by using Type = "Unknown". Refer to OpenTravel Code List Error Warning Type (EWT).
@Status	0..1	If present, recommended values are those enumerated in the OTA_ErrorRS, (NotProcessed Incomplete Complete Unknown) however, the data type is designated as string data, recognizing that trading partners may identify additional status conditions not included in the enumeration.
@ShortText	1	An abbreviated version of the error in textual format.
@Code	0..1	If present, this refers to a table of coded values exchanged between applications to identify errors or warnings. Refer to OpenTravel Code List Error Codes (ERR).

Element @Attribute	Num	Description/Contents
HTNG_ChargePostingRS / PostingGUID	1	A unique identifier for this posting.
HTNG_ChargePostingRS / TotalPostingAmount	1	An amount pertaining to the overall posting.
@Amount	1	The sum of all line items, taxes, gratuities, etc.
@CurrencyCode	0..1	The code specifying a monetary unit. Use ISO 4217, three alpha code.
@DecimalPlaces	0..1	Indicates the number of decimal places for a particular currency. This is equivalent to the ISO 4217 standard "minor unit." Typically used when the amount provided includes the minor unit of currency without a decimal point (e.g., USD 8500 needs DecimalPlaces="2" to represent \$85).
HTNG_ChargePostingRS / AuthorizationResult	0..1	Result information from the loyalty authorization process.
@AuthorizationCode	0..1	The unique code returned by the authorizing party. This is returned for successful authorizations.
@ApprovalDateTime	0..1	The date and time that the approval was issued.
@Result	1	The date and time that the approval was issued. Valid values are: <ul style="list-style-type: none"> • Approved • ApprovedWithPositiveID • Denied
HTNG_ChargePostingRS / AuthorizationResult / AccountAuthorization	0..1	Specifies account information about the customer seeking authorization.
@Amount	1	The number of units for which the authorization is good. For instance, the total number of points/miles needed to cover the transaction.
NonISO_CurrencyCode	1	Valid values are: <ul style="list-style-type: none"> • Points • Miles
HTNG_ChargePostingRS / Accounts	0	A collection of accounts that had charges posted to them.

Element @Attribute	Num	Description/Contents
HTNG_ChargePostingRS / Accounts / Account	1..n	The account for which a charge was applied.
@Type	0..1	The type of account. Possible values are "Account", "Room", "Extension".
@ID	1	An identifier representing the account for which an amount will be posted.
HTNG_ChargePostingRS / Accounts / LoyaltyInfo	0..1	Loyalty account information.
@PointBalance	1	The current balance of the loyalty account.
HTNG_ChargePostingRS / Vouchers	0..1	A collection of vouchers.
HTNG_ChargePostingRS / Vouchers / Voucher	1..n	Details of a paper or electronic document indicating prepayment.
@SeriesCode	0..1	Identification of a series of coupons or vouchers identified by serial number(s).
@SupplierIdentifier	0..1	Unique identifier of the electronic voucher, created by the supplier.
@Identifier	1	Unique identifier of the electronic voucher.
@Amount	1	The remaining balance of the voucher.
@CurrencyCode	1	The code specifying a monetary unit. Use ISO 4217, three alpha code.

4.3 Check Detail Lookup

4.3.1 Data Element Table – Request

Element @Attribute	Num	Description/Contents
HTNG_CheckDisplayRQ	1	Root element of the message.
@EchoToken	1	A reference for additional message identification, assigned by the requesting host system. When a request message includes an echo token, the corresponding response message MUST include an echo token with an identical value.

Element @Attribute	Num	Description/Contents
@TimeStamp	1	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).
@Version	1	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.
HTNG_CheckDisplayRQ / POS / Source	1	This holds details regarding the requestor. It may be repeated to also accommodate the delivery systems.
HTNG_CheckDisplayRQ / POS / Source / RequestorID	1	An identifier of the entity making the request (e.g., ATA/IATA/ID number, Electronic Reservation Service Provider (ERSP), Association of British Travel Agents (ABTA)).
@Type	0..1	A reference to the type of object defined by the UniqueID element. Refer to OpenTravel Code List Unique ID Type (UIT).
@ID_Context	0..1	Used to identify the source of the identifier (e.g., IATA, ABTA).
@ID	1	A unique identifying value assigned by the creating system. The ID attribute may be used to reference a primary-key value within a database or in a particular implementation.
HTNG_CheckDisplayRQ / CheckZoom	1	A container element used to convey the query parameters for performing a check display.
@ZoomInKey	1	A unique reference to a previously posted check.
@TextFormat	0..1	The format for which the check display should be represented. Possible values are "PlainText" and "HTML".

4.3.2 Data Element Table – Response

Element @Attribute	Num	Description/Contents
HTNG_CheckDisplayRS	1	Root element of the message.

Element @Attribute	Num	Description/Contents
@EchoToken	1	A reference for additional message identification, assigned by the requesting host system. When a request message includes an echo token, the corresponding response message MUST include an echo token with an identical value.
@TimeStamp	1	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).
@Version	1	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.
@Target	0..1	Used to indicate whether the request is for the Test or Production system.
HTNG_CheckDisplayRS / Success	0..1	The presence of the empty Success element explicitly indicates that the OpenTravel versioned message succeeded.
HTNG_CheckDisplayRS / Warnings	0..1	Used in conjunction with the Success element to define one or more business errors.
HTNG_CheckDisplayRS / Warnings / Warning	1..n	Used when a message has been successfully processed to report any warnings or business errors that occurred.
@Type	1	The Warning element MUST contain the Type attribute that uses a recommended set of values to indicate the warning type. The validating XSD can expect to accept values that it has NOT been explicitly coded for and process them by using Type ="Unknown". Refer to OpenTravel Code List Error Warning Type (EWT).
@Status	0..1	If present, recommended values are those enumerated in the OTA_ErrorRS, (NotProcessed Incomplete Complete Unknown) however, the data type is designated as string data, recognizing that trading partners may identify additional status conditions not included in the enumeration.
@ShortText	1	An abbreviated version of the error in textual format.

Element @Attribute	Num	Description/Contents
@Code	0..1	If present, this refers to a table of coded values exchanged between applications to identify errors or warnings. Refer to OpenTravel Code List Error Codes (ERR).
HTNG_CheckDisplayRS / CheckZoom	0..1	Contains the preformatted check representation.
@ZoomInKey	0..1	A unique reference to a previously posted check.
@TextFormat	0..1	The format for which the check display should be represented.

4.4 Synchronize Cache – Real-time/ Synchronize Cache – Push

4.4.1 Data Element Table – Request

Element @Attribute	Num	Description/Contents
HTNG_HotelCheckInNotifRQ	1	Root element of the message.
@EchoToken	1	A reference for additional message identification, assigned by the requesting host system. When a request message includes an echo token, the corresponding response message MUST include an echo token with an identical value.
@TimeStamp	1	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).
@Version	1	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.
HTNG_HotelCheckInNotifRQ / PropertyInfo	0..1	Information pertaining to a given hotel property.
@ChainCode	0..1	Describes whether the line item refers to a Chain (e.g., Courtyard, Doubletree).
@HotelName	0..1	The name of the Hotel.
@BrandCode	0..1	Describes whether the line item refers to a Brand (e.g., Marriott, Hilton).
@HotelCode	1	Describes whether the line item refers to an individual Hotel property.

Element @Attribute	Num	Description/Contents
HTNG_HotelCheckInNotifRQ / AffectedGuests	1	A collection of guests and/or reservations that are being checked into a room.
HTNG_HotelCheckInNotifRQ / AffectedGuests / UniqueID	1..n	An identifier used to uniquely reference an object in a system (e.g., an airline reservation reference, customer profile reference, booking confirmation number, or a reference to a previous availability quote).
@Type	0..1	A reference to the type of object defined by the UniqueID element. Refer to OpenTravel Code List Unique ID Type (UIT).
@ID	1	A unique identifying value assigned by the creating system. The ID attribute may be used to reference a primary-key value within a database or in a particular implementation.
HTNG_HotelCheckInNotifRQ / Room	0..1	Used to convey information about a single room or a suite comprised of room components.
@RoomID	0..1	The room number.
HTNG_HotelCheckInNotifRQ / Room / RoomType	1	
@InvBlockCode	0..1	The group code.
HTNG_HotelCheckInNotifRQ / Room / TelephoneExtensions	0..1	A collection of telephone extensions.
HTNG_HotelCheckInNotifRQ / Room / TelephoneExtensions / TelephoneExtension	1..n	A telephone extension that is part of a single room or a suite of component rooms.
HTNG_HotelCheckInNotifRQ / HotelReservations	0..1	Used to convey information about the arriving reservation and its supporting guest information.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation	1..n	One line in a list of hotel reservations. It may contain the reservation ID, booked date, arrival and departure dates, number of nights and rooms, hotel info, guest info, and room info
@ResStatus	1	Indicates the status of the reservation.
@RoomStayReservation	0..1	Boolean is True if this reservation is reserving rooms; False if it is only reserving services.

Element @Attribute	Num	Description/Contents
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / UniqueID	1..5	An identifier used to uniquely reference an object in a system (e.g., an airline reservation reference, customer profile reference, booking confirmation number, or a reference to a previous availability quote).
@Type	0..1	A reference to the type of object defined by the UniqueID element. Refer to OpenTravel Code List Unique ID Type (UIT).
@ID	1	A unique identifying value assigned by the creating system. The ID attribute may be used to reference a primary-key value within a database or in a particular implementation.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / RoomStays	0..1	Collection of room stays.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / RoomStays / RoomStay	1..n	Details on the Room Stay including Guest Counts, Time Span of this Room Stay, pointers to Res Guests, guest Memberships, Comments and Special Requests.
@MarketCode	0..1	The code that relates to the market being sold to (e.g., the corporate market, packages).
@DiscountCode	0..1	A discount code known to the hotel.
@RoomStayStatus	1	Identifies the status of the room stay.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / RoomStays / RoomStay / RoomRates	0..1	A collection of Room Rates associated with a particular Room Stay.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / RoomStays / RoomStay / RoomRates / RoomRate	i..n	Individual rate amount. This rate is valid for a range of number of occupants and an occupant type.
@RoomTypeCode	0..1	Specific system room type code, ex: A1K, A1Q, etc.

Element @Attribute	Num	Description/Contents
@RatePlanCode	0..1	A string value may be used to request a particular code or an ID if the guest qualifies for a specific rate, such as AARP, AAA, a corporate rate, etc., or to specify a negotiated code as a result of a negotiated rate.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / RoomStays / RoomStay / RoomRates / RoomRate / GuestCounts	0..1	A collection of Guest Counts associated with the room rate.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / RoomStays / RoomStay / RoomRates / RoomRate / GuestCounts / GuestCount	1..n	A recurring element that identifies the number of guests and ages of the guests.
@AgeQualifyingCode	0..1	A code representing a business rule that determines the charges for a guest based upon age range (e.g., Adult, Child, Senior, Child With Adult, Child Without Adult). This attribute allows for an increase in rate by occupant class. Refer to OpenTravel Code List Age Qualifying Code (AQC).
@Count	0..1	The number of guests in one AgeQualifyingCode or Count.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / RoomStays / RoomStay / TimeSpan	0..1	The Time Span which covers the Room Stay.
@Start	1	The starting value of the time span.
@End	1	The ending value of the time span.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / ResGuests	0..1	Collection of guests associated with the reservation. The ResGuest object contains the information about a guest associated with a reservation.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / ResGuests / ResGuest	1..n	

Element @Attribute	Num	Description/Contents
@VIP	0..1	Guest is VIP indicator.
@PrimaryIndicator	0..1	When true indicates this is the primary guest.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / ResGuests / ResGuest / Profiles	1	A collection of Profile objects or Unique IDs of Profiles.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo	1..n	A collection of Profiles or Unique IDs of Profiles.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo / UniqueID	1	A unique ID for a Profile. This element repeats to accommodate multiple unique IDs for a single Profile across multiple systems.
@Type	1	A reference to the type of object defined by the UniqueID element. Refer to OpenTravel Code List Unique ID Type (UIT).
@ID	1	A unique identifying value assigned by the creating system. The ID attribute may be used to reference a primary-key value within a database or in a particular implementation.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo / Profile	0..1	Provides detailed information regarding either a company or a customer Profile.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo / Profile / Customer	0..1	Detailed customer information for this Profile.

Element @Attribute	Num	Description/Contents
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo / Profile / Customer / PersonName	1	Detailed name information for the customer.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo / Profile / Customer / PersonName / NamePrefix	0..1	The Salutation for the name. This SHOULD be a value representing an individual (Mr., Mrs., Dr.) and not Mr. and Mrs..
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo / Profile / Customer / PersonName / GivenName	0..1	Given name, first name or names.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo / Profile / Customer / PersonName / SurName	0..1	Family name, last name. May also be used for full name if the sending system does not have the ability to separate a full name into its parts, e.g. the surname element may be used to pass the full name.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / ResGuests / ResGuest / Profiles / ProfileInfo / Profile / Customer / CustLoyalty	0..1	Loyalty program information for the customer.
@MembershipID	1	Unique identifier of the member in the program (membership number, account number, etc.).
@VendorCode	0..1	Indicate the partner(s)/vendor(s) for which the customer loyalty number is valid.
@ProgramID	0..1	Identifier to indicate the company owner of the loyalty program.
@LoyalLevel	0..1	Indicates special privileges in program assigned to individual.

Element @Attribute	Num	Description/Contents
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / ResGuests / ResGuest / Comments	0..1	A collection of Comment objects. Comments which apply to the whole Reservation or a particular Room Stay or Service.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / ResGuests / ResGuest / Comments / Comment	1..n	Comment details.
@Name	0..1	In many cases the description repeats, this will allow you to define the information that is being sent, typically used when multiple Comment elements are being sent.
@GuestViewable	0..1	When true, the comment may be shown to the consumer. When false, the comment may not be shown to the consumer.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / ResGuests / ResGuest / Comments / Comment / Text	1	Formatted text content.
@Language	0..1	Language identification.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / TPA_Extensions	0..1	A placeholder in the schema to allow for additional elements and attributes to be included if required, per Trading Partner Agreement (TPA).
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / TPA_Extensions / TPA_Extension	1..n	A single instance of a Trading Partner Agreement extension.
HTNG_HotelCheckInNotifRQ / HotelReservations / HotelReservation / TPA_Extensions / TPA_Extension / AvailableCredit	0..1	Used to convey information about the available credit of an account.
@CurrencyCode	0..1	The code specifying a monetary unit. Use ISO 4217, three alpha code.

Element @Attribute	Num	Description/Contents
@Amount	1	The total amount of available credit as determined by the responding system.
@DecimalPlaces	0..1	Indicates the number of decimal places for a particular currency. This is equivalent to the ISO 4217 standard "minor unit." Typically used when the amount provided includes the minor unit of currency without a decimal point (e.g., USD 8500 needs DecimalPlaces="2" to represent \$85).

4.4.2 Data Element Table – Response

Element @Attribute	Num	Description/Contents
HTNG_HotelCheckInNotifRS	1	Root element of the message.
@EchoToken	1	A reference for additional message identification, assigned by the requesting host system. When a request message includes an echo token, the corresponding response message MUST include an echo token with an identical value.
@TimeStamp	1	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).
@Version	1	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.
@Target	0..1	Used to indicate whether the request is for the Test or Production system.
HTNG_HotelCheckInNotifRS / Success	0..1	The presence of the empty Success element explicitly indicates that the OpenTravel versioned message succeeded.
HTNG_HotelCheckInNotifRS / Warnings	0..1	Used in conjunction with the Success element to define one or more business errors.
HTNG_HotelCheckInNotifRS / Warnings / Warning	1..n	Used when a message has been successfully processed to report any warnings or business errors that occurred.

Element @Attribute	Num	Description/Contents
@Type	1	The Warning element MUST contain the Type attribute that uses a recommended set of values to indicate the warning type. The validating XSD can expect to accept values that it has NOT been explicitly coded for and process them by using Type = "Unknown". Refer to OpenTravel Code List Error Warning Type (EWT).
@Status	0..1	If present, recommended values are those enumerated in the OTA_ErrorRS, (NotProcessed Incomplete Complete Unknown) however, the data type is designated as string data, recognizing that trading partners may identify additional status conditions not included in the enumeration.
@ShortText	1	An abbreviated version of the error in textual format.
@Code	0..1	If present, this refers to a table of coded values exchanged between applications to identify errors or warnings. Refer to OpenTravel Code List Error Codes (ERR).

4.5 Offline Status Notification

4.5.1 Data Element Table – Request

Element @Attribute	Num	Description/Contents
HTNG_SystemStatusNotifRQ	1	Root element of the message.
@EchoToken	1	A reference for additional message identification, assigned by the requesting host system. When a request message includes an echo token, the corresponding response message MUST include an echo token with an identical value.
@TimeStamp	1	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).
@Version	1	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.

Element @Attribute	Num	Description/Contents
HTNG_SystemStatusNotifRQ / SystemStatus	1	The container element used to convey information about a change in system status.
@Status	1	“Online” or “Offline.”
@ScheduledReturn	0..1	The anticipated time the system will return to an on-line status.

4.5.2 Data Element Table – Response

Element @Attribute	Num	Description/Contents
HTNG_SystemStatusNotifRS	1	Root element of the message.
@EchoToken	1	A reference for additional message identification, assigned by the requesting host system. When a request message includes an echo token, the corresponding response message MUST include an echo token with an identical value.
@TimeStamp	1	Indicates the creation date and time of the message in UTC using the following format specified by ISO 8601; YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 pm UTC becomes 2003-11-20T13:59:38Z).
@Version	1	For all OpenTravel versioned messages, the version of the message is indicated by a decimal value.
@Target	0..1	Used to indicate whether the request is for the Test or Production system.
HTNG_SystemStatusNotifRS / Success	0..1	The presence of the empty Success element explicitly indicates that the OpenTravel versioned message succeeded.
HTNG_SystemStatusNotifRS / Warnings	0..1	Used in conjunction with the Success element to define one or more business errors.
HTNG_SystemStatusNotifRS / Warnings / Warning	1..n	Used when a message has been successfully processed to report any warnings or business errors that occurred.

Element @Attribute	Num	Description/Contents
@Type	1	The Warning element MUST contain the Type attribute that uses a recommended set of values to indicate the warning type. The validating XSD can expect to accept values that it has NOT been explicitly coded for and process them by using Type ="Unknown". Refer to OpenTravel Code List Error Warning Type (EWT).
@Status	0..1	If present, recommended values are those enumerated in the OTA_ErrorRS, (NotProcessed Incomplete Complete Unknown) however, the data type is designated as string data, recognizing that trading partners may identify additional status conditions not included in the enumeration.
@ShortText	1	An abbreviated version of the error in textual format.
@Code	0..1	If present, this refers to a table of coded values exchanged between applications to identify errors or warnings. Refer to OpenTravel Code List Error Codes (ERR).

5 Appendices

5.1 Terms

Account Lookup Requester	A system that needs to determine whether a particular guest is staying in the hotel, or to look up an existing (house, membership, etc.) account.
Account Lookup Responder	A system that holds information about hotel guests, their associated reservations and accounts.
Account Posting Requester	A system that has the need to post charges to guest rooms or accounts.
Account Posting Responder	A system that houses guest and account information; receives charge requests and posts them to the appropriate folios.
Cache Holder	A system that caches a view of (current, arriving or departing) hotel guests and accounts for the purpose of being able to perform local lookups in case communications between systems becomes unavailable.
Check Detail Lookup Requester	A system that has previously received a charge posting but would like to view the charge detail.
Check Detail Lookup Responder	A system that had previously posted a charge to another system and is able to present a full representation of the original charge.
Check Zoom	A key which can be used by the receiving system to acquire additional details for the posting; this is referred to as the “Zoom In Key” in the HTNG Single Guest Itinerary spec.
Customer Loyalty Requester	A system that needs to: <ul style="list-style-type: none">• Settle against a loyalty point account.• Be able to request a new customer loyalty program member enrollment be created.• Be able to request a new customer loyalty program member enrollment be created.
Customer Loyalty Responder	A system that manages customer loyalty program memberships and validates transactions.

Data Source Provider	A system that publishes changes to its data to another system.
Offline Status Publisher	A system that notifies another system that it is about to become unavailable and that offline or manual procedures should be taken until system communication resumes.
Offline Status Subscriber	A system that receives a notification from another system that communication is about to become unavailable.
Summary Data Posting	A charge posting that contains summarized financial information from multiple transactions. Because the posting is composed of summary information it will not contain a guest check or transaction number from the originating system.

5.2 Implementation Notes

5.2.1 Void versus Refund

Trading partners will need to decide whether they will use Void Posting, Charge Posting (with negative values) or a combination of both.

For clarity, the Void Posting should be used when a previous posting must be reversed in its entirety due to a posting error. This is important for complex accounting scenarios that require charge routing, package entitlements (and breakage), loyalty benefits, etc., where the Account Posting Requester may not be aware of the business rules of the Account Posting Responder.

Care should be taken to ensure proper controls and logging is in place to prevent theft.

5.2.2 Financial Reconciliation

It is possible to balance financial information between POS and PMS systems using the financial reconciliation payload. So that totals can balance, it is assumed that reconciliation will occur for (1) all data that was previously transmitted for the time period or business day in question, and (2) only previously transmitted data will be included in the totals that are reconciled.

The primary reason that this functionality is provided is so that the night audit process can be automated, saving time for manual procedures that occur during audit when the systems balance, and to validate data integrity of the interface. An added benefit is that this functionality can also be used to help test that a recently implemented interface has been configured properly. For audit purposes, it is possible to balance to a subset of data based on how the split code/mapping tables are set up for the interface, and financial reconciliation

could occur for (a) transactions codes and/or (b) tender types. This allows for (a) revenue audit automation and/or (b) cash control/cash reconciliation.

In the case where systems don't balance, it will be possible for a PMS to automatically generate additional reports to help identify and correct manually the out-of-balance condition, or to notify the system, an auditor, or the implementation technician about the condition. In the case of cloud-based systems with many interfaces to many properties, it is possible to ensure proper data integrity between each interface instance in an automated manner.

There are several reasons that POS and PMS financial information may not balance. There is no specification in this document that outlines how to handle or process the data that is sent via the financial reconciliation payload. It is up to the PMS developer to determine how to handle differences in financial information between systems based on the data transmitted from the Financial Reconciliation payload.

Differences in totals could occur for the following reasons, plus others that are not anticipated in this document:

- Not all transactions have posted (There may have been an interface failure, or offline transactions may not have posted.)
- Not all transactions that were posted have been processed (timing issues with data being compared)
- Rounding differences may be present between the two systems due to differences in data processing rules
- Interface setup tables don't match between systems

It is up to the developer to set up acceptable tolerances for variance, similar to financial accounting practices related to materiality, or per other business rules that they see fit.

5.2.3 Account Posting Payload

It is possible to post both detailed and summary transactions in an account posting payload. Summary transactions aggregate data from multiple transactions in a POS system, and therefore do not have a unique check number that can tie back to a single POS transaction record or closed check. This allows for systems to post financial information for revenue reporting or audit purposes that might not otherwise be transmitted because it is not needed for posting to a guest folio. A secondary benefit is that detailed transaction data duplication can be significantly reduced if summary transactions are posted when detailed transaction postings are not required.

5.2.4 Summary Transaction Posting

The summary transaction posting feature is designed to aggregate data from multiple transactions for posting a transaction once at the end of a business day or reporting period, to reduce interface traffic where transaction detail is not needed, or to use the PMS interface to post transactions to a back office system to which transactions could not otherwise be electronically posted. Summary transactions are either not typically intended for posting to a guest or membership account, or backup transaction information is not kept electronically in the case of manual systems. Therefore, there is not normally a single “check” or viewable guest receipt associated with the transaction in electronic format, and the Check Zoom data field would not be populated for the transaction.

NOTE: Previously posted transactions (e.g. Room Charges) should never be included in a summary transaction posting as it may result in double postings or accounting systems may be out of balance!

5.2.4.1 Transaction Mapping

It is possible that summary transactions are intended to be posted to a revenue center or account combination different from the system posting the transaction. For example, if a summary transaction for Audio Visual revenue is posted from a restaurant POS system, it might be most appropriate or desirable for the revenue to show up in the Banquet department’s P&L statement (a different revenue center and account combination in the back office or G/L system). However, typical POS system and interface configurations would post transactions originating from a given terminal or revenue center to the P&L associated with the terminal or revenue center. Although solutions might vary to work around this, most POS systems will have a means for setting up an alternate revenue center, revenue category, or item classification that will allow the transaction to be posted properly by the PMS once the transaction is mapped properly in the PMS interface configuration tables. The POS terminal might be placed in a different revenue center temporarily (i.e. a revenue center set up for Banquets), or items with a different revenue category (i.e. A/V Sales) and/or revenue classification (i.e. Equipment Rental) might be set up that are associated with a mapping entry in the PMS interface table that will post the transaction to the proper account. In this scenario, the POS system and the PMS interface configuration tables must be properly configured for the summary transactions to post properly.

5.2.4.2 Transaction Examples

- Post revenue from a manual operation via a POS system (i.e. post A/V Sales via a POS system)
- Summarize cash transactions by cashier for posting to a cash control system (e.g. when it is not desirable to post all cash transactions via the interface, but cash control information is needed)
- Post quarterly member account charges from a manual ledger (i.e. condo repair fees at a condo hotel property)
- If using a PMS for posting of accounting data, summary transactions can be used to post financial information for tender types and revenue categories that are not typically posted:
 - Net sales by revenue category
 - Sales with inclusive taxes by revenue category
 - Sales received by each payment type
 - Dollar amount of each comp
 - Dollar amount of each promo/discount
 - Dollar amount of each tax
 - Number of transactions for the specified day

5.2.5 Best Practices for Data Mining

To aid in collection and analysis of actionable customer information, many PMSs (or Enterprise Back Office systems implementing this interface) will want to have as much customer information as possible to extract from transaction postings. Some POS systems may not be able to associate a check with a customer profile in a PMS (for instance in the case of cash-paying guests) or be able to correlate customer information with other POS systems, but data mining applications or processes can correlate this data if it is made available. To help with data analysis and data mining processes, whenever possible, a unique customer profile identifier should be included in a transaction posting. In certain cases, other information may be used to identify a unique guest (e.g. name, room number, account number), or other items that might be used to correlate transactions posted from other systems. For example, a table number, check open time and menu item details from a cash restaurant check posting could be matched to a table number, seating time, guest name, and e-mail address from a table management system. This information could then be used by marketing staff to create and send a customized offer to that guest based on the information gleaned from the transaction data.

Also, it may be desirable to post additional tender types that contain guest information to a PMS or Enterprise Back Office system where these transactions have not historically been posted in the past. For example, if cash or credit card transactions are posted and can be correlated with guest information it would be possible to more accurately understand total spend per guest, something that has typically been difficult to do.

5.3 Referenced Documents

The following table shows the documents upon which this document depends:

Document Title	Location/URL
HTNG Customer Profile v3.0 Specification	http://collaboration.htng.org/specs/documents.php?action=show&dcat=48&gdid=25350
HTNG Folio Detail Exchange v2.0 Specification	http://collaboration.htng.org/specs/documents.php?action=show&dcat=48&gdid=25349
HTNG Guest & Room Status Messaging Specification v2.0	http://collaboration.htng.org/specs/documents.php?action=show&dcat=42&gdid=23699
HTNG Protocol & Message Transport Event Notification Specification v2.0	https://collaboration.htng.org/specs/documents.php?action=show&dcat=54&gdid=26570
HTNG Single Guest Itinerary v2.0	http://collaboration.htng.org/specs/documents.php?action=show&dcat=25&gdid=22364
OpenTravel Alliance Specifications	http://opentravel.org/Specifications/Default.aspx