Property Web Services Specifications
Release 2008A

Issue 4 includes an addendum to Single Guest Itinerary (Part 2), and a minor update to the 'About HTNG' section.

March 31, 2008

Acknowledgements

Many companies and individuals contributed significant time and effort the development of these specifications. However, their work was greatly facilitated by the willingness of a few companies to contribute existing specifications as a starting point.

HTNG gratefully acknowledges significant contributions of pre-existing work by the following companies, in support of specific portions of these specifications.

Property Web Services Workgroup

About HTNG

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

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

In June 2005, HTNG announced the first-ever "Branding and Certification Program" for hotel technology. This program will enable vendors to certify their products against open HTNG specifications, and to use the "HTNG Certified" logo in their advertising and collateral materials.

It will enable hotels to determine which vendors have completed certification of their products against which specific capabilities, and the environments in which performance is certified. HTNG's vision is to achieve a flexible technical environment that will allow multiple vendors' systems to interoperate and that will facilitate vendor alliances and the consolidation of applications, in order to provide hotels with easily managed, continually evolving, cost-effective solutions to meet their complete technology needs on a global basis.

Document History

Issue 1	March 31, 2008	Initial Release
Issue 2	June 12, 2008	Added note to Section 14.2: FetchCheckDetails to indicate this is not intended to part of this version of the spec
Issue 3	November 24, 2008	Revised Part 3: Digital Signage as incorrect version was included in original publication
Issue 4	May 27, 2010	Updated 'About HTNG' Section, added Addendum to the Single Guest Itinerary Specification

Table of Contents

Chapter 1 Introduction

This document is Release 2008A of the HTNG Property Web Services Specifications.

For convenience, these are presented as a single multi-part document, with each part comprising a different specification.

The 2008A Release comprises the following parts:

1. Web Services Framework
   Version 2.1

2. Single Guest Itinerary
   Version 1.0.1

3. Digital Signage
   Version 1.0

4. Open Data Exchange
   Version 1.0

5. Back Office Integration
   Version 1.0

6. Guest Self Service
   Version 1.0

1.1 Specification Status

Of the specification included in this document, the first three – Web Services Framework 2.1, Single Guest Itinerary 1.0, Digital Signage 1.0 - are included in the HTNG Release 2008A Certification Program.

The last three specifications - Open Data Exchange 1.0, Back Office Integration 1.0, Guest Self Service 1.0 - are not yet included in the Certification Program, but it is expected that they will be included in a future release of the Certification Program. **Note: These specifications may change prior to their inclusion in a future release of the Certification Program.**

Note: Single Guest Itinerary 1.0 is unchanged from the version published as part of the HTNG Release 2007a Certification Program.

Part 1 ***Web Services Framework Version 2.1***

Chapter 2 Document Information

2.1 Overview

A key group of technologists from across the industry were formed into a workgroup to revise the methodology for system interconnection via Web Services.

This group included the following people representing the following companies:-

Name	Company Represented	Email
Kristofer Agren	OpenCourse Solutions	kagren@opencourse.com
Sophie Grigg	PAR Springer-Miller Systems	sophie_grigg@springermiller.com
Alex Lobakov	PAR Springer-Miller Systems	alex_lobakov@springermiller.com
Alex Shore	Newmarket International	AShore@newmarketinc.com
Tom Gresham	MICROS Systems, Inc.	tgresham@micros.com
Andreas Hagedorn	Trust International	ahagedorn@trustinternational.com
Brad More	Theodatus	brad.more@theodatus.com
Mark Pullen	InfoGenesis	mpullen@infogenesis.com
Doug Rice	HTNG	douglas.rice@htng.org

The group met together during 2006 & 2007 to prepare this specification material. During Hitec (June 2006) various vendors were able to demonstrate the use of the new Framework 2.1 methodology to interface systems for various HTNG initiatives. In addition a number of workgroups have either used this Framework as their connectivity mechanism or plan to do so.

This specification describes the Framework 2.1 methodology it remains a working document until a number of vendors are live with Framework 2.1.

2.2 Referenced Documents

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

Name	Location
SOAP 1.1	
SOAP 1.2	
Web Services Addressing 1.0 - Core W3C Working Draft: 2006-05-09	http://www.w3.org/TR/2006/REC-ws-addr-core-20060509

Chapter 3 Overview

3.1 Overview

This specification outlines a set of existing open standards, patterns and practices that have gained significant acceptance throughout the IT industry that must be supported by an implementer to claim HTNG Framework 2.1 compliance. The framework prescribes a service oriented architecture implemented using Web Services.

An HTNG Framework 2.1 compliant Web Service MUST adhere to the following:

1. Support both SOAP 1.1 and SOAP 1.2
2. Be expressed fully in WSDL and XML Schema
3. Support WS-Addressing (Web Services Addressing 1.0 - Core W3C Working Draft: 2006-05-09 - http://www.w3.org/TR/2006/REC-ws-addr-core-20060509)
4. Be accessible using synchronous and/or asynchronous HTTP as described below
5. There is an assumption that you are processing XML messages correctly utilizing best practices and known conventions

The following is RECOMMENDED

1. Usage of WS-Security to authenticate messages and secure message content. Only the WS-Security 1.0 specification is supported within the context of this specification.
2. Usage of asynchronous HTTP as described below
3. Usage of SOAP faults as described below
4. Usage of the XML Schema and WSDL construction best practices as described below
5. Vendors conform with v1.1 of the WS-I Basic Profile specification - http://www.ws-i.org/Profiles/BasicProfile-1.1.html

3.2 WS Addressing

The group reviewed many of the specifications in place and in use across many industries. It was felt that the use of existing standards was the optimal way to create the most effective communication standard. This documentation uses the standards described in the Web Services section of the http://www.w3.org

The group decided that guidelines defined within Web Services Addressing were appropriate for Framework 2.1. The W3C Working Draft of May 2006 (2006-05-09 - http://www.w3.org/TR/2006/REC-ws-addr-core-20060509) was chosen as the WS-Addressing version to use because it provided a platform that was well supported by many of the common software platforms the members were using to develop their software.

Synchronous and Asynchronous communications are in use actively by many HTNG members using the existing HTNG header. Framework 2.1 provides support for both communication patterns. In addition we have provided guidelines that should make the communication process more robust and standardized.

3.3 HTTP Communication Patterns

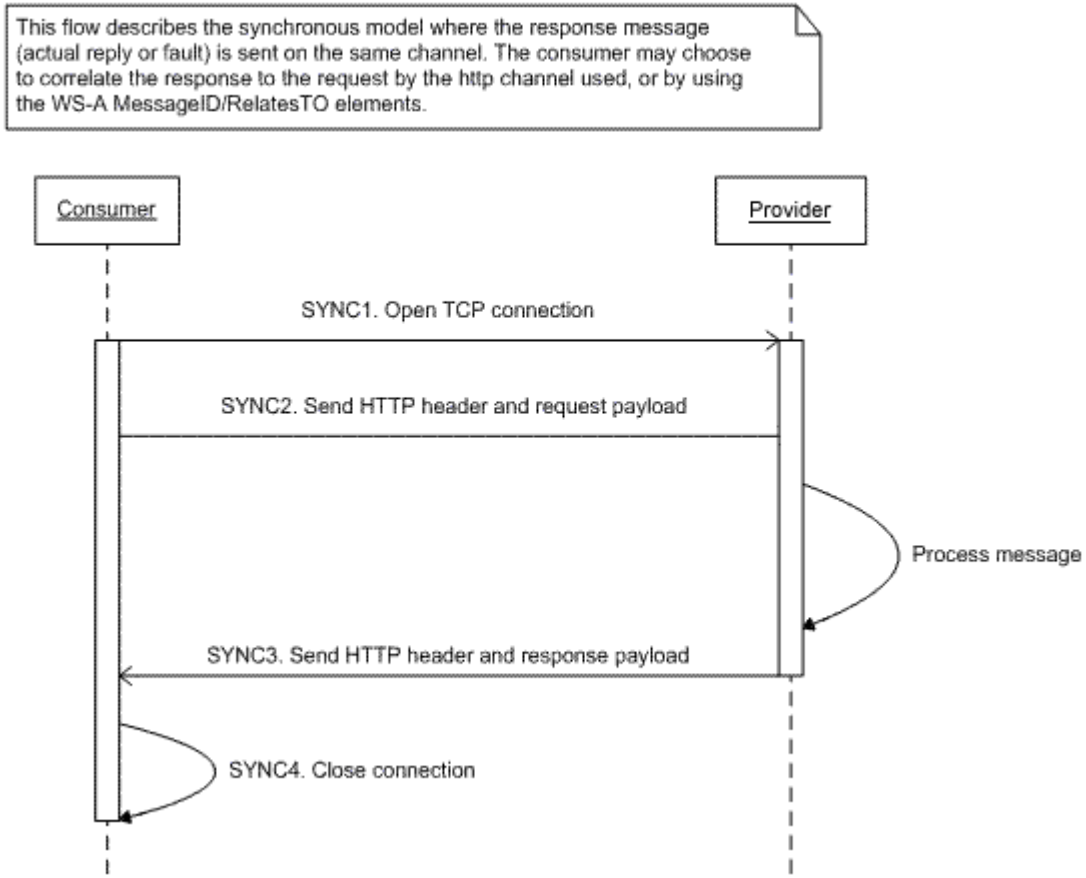
The synchronous pattern, whilst straightforward, is included here for reference and to aid in communicating a best practice approach. As different web service frameworks handle asynchronous communication differently, the approach described in the 'Async pattern' section is recommended. Whilst an implementer may choose to support other async patterns, at the very least this async pattern must be supported in order to claim HTNG compliance. The full details of asynchronous communication and the associated fault handling is detailed within this specification along with some sample applications that demonstrate an implementation. Synchronous communications are defined in Chapter Chapter 4. Asynchronous communications are defined in Chapter Chapter 5.

Chapter 4 The Synchronous Communication Process

Synchronous communication is defined as a request/response over the same channel. The overall principals in this communication are defined as follows:-

- Each message will supply a unique MessageID. Should the same message be transmitted with a different MessageID (many organizations utilize a GUID for this purpose), this will be treated as a separate message by the recipient.
- If reply to element contains the anonymous URI then the response will be provided on the same channel, by definition this becomes a Synchronous transaction. Please see http://www.w3.org/TR/2006/REC-ws-addr-core-20060509 (Section 3 – Message Addressing Properties)
-

Diagrammatically this can be represented via the following sequence diagram:



Note: Examples provided in the specification are from the original Framework 2.0 specification - 1.0.8, as such the soap Envelope references the 2004-08 namespace. This has been maintained in the example(s) for backward compatibility.

4.1 SYNC1

The caller sends the request message, and indicates that the response should be sent on the same connection by supplying the WS-Addressing “anonymous” URI in the wsa:ReplyTo element. Note that if the anonymous URI is used and there is both a wsa:ReplyTo and wsa:FaultTo element, both wsa:ReplyTo and wsa:FaultTo elements MUST use the anonymous URI, i.e. it is not possible to use the anonymous URI for wsa:ReplyTo but not wsa:FaultTo, or vice versa.

4.1.1 Sync - Sample message with HTTP header

```
POST /MyService.asmx HTTP/1.1
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
Connection: close
SOAPAction: http://xyz/MyService/SayHello

(01) <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
            xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing";>
(02)     <soap:Header>
(04)         <wsa:MessageID>uuid:214A50B2-E62E-4f8b-BD97-62ABE31E15C2</wsa:MessageID>
(05)         <wsa:ReplyTo>
(06)             <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
(07)         </wsa:ReplyTo>
(08)         <wsa:To>http://xyz/MyService</wsa:To>
(09)         <wsa:Action>http://xyz/MyService/SayHello</wsa:Action>
(10)     </soap:Header>
(11)     <soap:Body>
(12)         <m:SayHello xmlns:m="http://xyz/MyService">
(13)             <m:MyName>John Doe</m:MyName>
(14)         </m:SayHello>
(15)     </soap:Body>
(16) </soap:Envelope>
```

4.2 SYNC2

The Web Service sends back the response on the same connection that the request came in on.

4.2.1 Sync - Sample message with HTTP response header

```
HTTP/1.1 200 OK
Date: Wed, 10 May 2006 11:30:07 GMT
Content-Length: nnnn
Content-Type: text/xml; charset="utf-8"

(01) <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
(02)     <soap:Header>
(03)         <wsa:MessageID>uuid:5ED743DD-C051-43e8-9287-D349DEAD38FB</wsa:MessageID>
(04)         <wsa:RelatesTo>uuid:214A50B2-E62E-4f8b-BD97-62ABE31E15C2</wsa:RelatesTo>
(05)         <wsa:To>http://abc:1234/MyClient </wsa:To>
(06)         <wsa:Action>http://xyz/MyService/SayHelloResponse</wsa:Action>
```

```
(07)     </soap:Header>
(08)     <soap:Body>
(09)         <m:SayHelloResponse xmlns:m="http://xyz/MyService">
(10)             <m:Greeting>Hello John Doe</m:Greeting>
(11)         </m:SayHelloResponse>
(12)     </soap:Body>
(13) </soap:Envelope>
```

4.2.2 Sync - Sample fault with HTTP response header

```
HTTP/1.1 200 OK
Date: Wed, 10 May 2006 11:30:07 GMT
Content-Length: nnnn
Content-Type: text/xml; charset="utf-8"

(01) <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
(02)     <soap:Header>
(03)         <wsa:MessageID>uuid:AF6F10EE-B2A8-4080-BEA0-0A5F03100C60</wsa:MessageID>
(04)         <wsa:RelatesTo>uuid:214A50B2-E62E-4f8b-BD97-62ABE31E15C2</wsa:RelatesTo>
(05)         <wsa:To>http://abc:1234/MyClient</wsa:To>
(06)         <wsa:Action>http://www.w3.org/2005/08/addressing/fault</wsa:Action>
(07)     </soap:Header>
(08)     <soap:Body>
(09)         <soap:Fault xmlns:m="http://xyz/MyService">
(10)             <faultcode>m:MyNameNotSet </faultcode>
(11)             <faultstring>No name was specified, unable to say hello</faultstring>
(12)             <detail>
(13)                 <m:MyNameNotSet>
(14)                     <m:SomeElement>Some additional information</m:SomeElement>
(15)                 </m:MyNameNotSet>
(16)             </detail>
(17)         </soap:Fault>
(18)     </soap:Body>
(19) </soap:Envelope>
```

4.3 Overall Fault Handling

SOAP Faults will be provided as a mechanism for handling error conditions. We highly recommend that faults are declared in the WSDL. These may include faults that are business dependent. For example, methods that create reservations may want to return failures for defined failure reasons like "missing arrival or departure date".
Since faults relating to transportation are not (typically) known ahead of time, these would not normally be declared in the WSDL.
A response that contains a fault should be sent to the same address as a reply would, unless a specific wsa:FaultTo element was specified in the header of the request message
Documentation describing an example of SOAP fault handling is attached in Appendix 2.

4.4 WSDL Construction and Change Management

We strongly recommended the use of HTTP for transporting messages. This specification primarily focuses on Web Services using HTTP as the transport medium.

4.4.1 Recommendations for Change Management for Documents and Specifications Involving Web Services

A workgroup is responsible for the WSDL and XML Schema that it delivers together with a specification. A new version of the specification results in a new WSDL and XML Schema with new namespaces. Since a new version will have a new namespace, no two versions are syntactically compatible. It is up to the workgroup to define a migration path (if one exists) in the specification.
It is recommended, but not required, that the target namespaces for the WSDL and XML Schema adhere to the following format:

- WSDL:http://htng.org/<workgroup acronym>/<year and target release>/<specification name>
- XML Schema: http://htng.org/<workgroup acronym>/<year and target release>/<specification name>/<schema name>/Types

For draft specifications, the following namespace format is recommended:

- WSDL: http://htng.org/<workgroup acronym>/<year and target release>/DRAFT/<specification name>
- XML Schema: http://htng.org/<workgroup acronym>/<year and target release>/DRAFT/<specification name>/<schema name>/Types

For example, if the Property Web Services Workgroup publishes a standard for Single Guest Itinerary for the 2008A release of the specification, the namespaces for the WSDL and XML Schema would be:

- WSDL: http://htng.org/PWS/2008A/SingleGuestItinerary
- XML Schema(s): http://htng.org/PWS/2008A/SingleGuestItinerary/Common/Types

These URIs both serve as a unique identifier for the namespace as well as the URLs for retrieving the WSDL and XML Schema.

4.4.1.1 Release Process

1. Release of draft specification. A draft specification is assumed to consist of at least three artifacts; a document explaining the specification, a WSDL file, and an XML Schema file. Before the release is made, the WSDL and XML Schema namespaces should be updated.
2. Review process
3. Changes to draft specification, if needed.
4. Ratification process
5. Release of ratified specification. Update namespace of WSDL and XSD, and upload WSDL and XML Schema to the htng.org site (if and when this is available) so that in the best case, it is possible to use the namespace URIs to retrieve the WSDL and XML Schema.

Once a specification has been ratified, it becomes static and any changes would need to be considered for the next version. Once it is determined that a new release should take place, a new target release is established (ie 2008A or 2008B), and the release process starts over from the beginning.

4.4.2 Change Management

Changes to a specification can be proposed by a vendor, either by submitting an extension (as described in the Extensibility section of this document) or by submitting a written request to HTNG. The workgroup responsible for the specification will periodically review any suggested changes, and if deemed to be of use to a wider audience, the changes are incorporated into a new version.
All extensions that are considered to become part of the standards will be assimilated into the standard XML Schema and will then live in the same namespace as the standard.

4.4.3 Extensibility

Once a specification has been ratified and released, it is likely that vendors implementing the specification would like to enhance the specification either by augmenting the message content or even adding completely new functionality.

4.4.3.1 For Writers of a Specification

It is recommended that the following guidelines are taken into account when building the XML Schema for a specification to ensure that the message content is extensible.
At the bottom of a complexType that should be extensible, use the construct and specify that other elements are allowed as long as they are from another namespace.
An Example follows:

```
(01)<?xml version="1.0" encoding="UTF-8"?>
(02)<xs:schema targetNamespace="http://htng.org/PWS/2008A/SomeSpecification/schema name/Types"
xmlns:tns="http://htng.org/PWS/2008A/SomeSpecification/schema name/Types"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
(03) <xs:element name="Customer">
(04)    <xs:complexType>
(05)      <xs:sequence>
(06)       <xs:element name="LastName" type="xs:string"/>
(07)       <xs:element name="FirstName" type="xs:string"/>
(08)       <xs:element name="Address" type="tns:AddressType"/>
(09)       <xs:any minOccurs="0" maxOccurs="unbounded" namespace="##other"/>
(10)      </xs:sequence>
(11)    </xs:complexType>
(12)  </xs:element>
(13)   <xs:complexType name="AddressType">
(14)      <xs:sequence>
(15)       <xs:element name="Street" type="xs:string"/>
(16)       <xs:element name="ZipCode" type="xs:string"/>
(17)       <xs:any minOccurs="0" maxOccurs="unbounded" namespace="##other"/>
(18)      </xs:sequence>
(19)   </xs:complexType>
(20)</xs:schema>
```

It is highly recommended that this pattern is used in as many (if not all) complexTypes that are defined in the XML Schema to allow for maximum extensibility.
NOTE: This model was chosen for its simplicity. There are other extensibility models that would work as well, but the <xs:any> version is straightforward and does not require dependency on some common XML Schema.

4.4.3.2 For Implementors of a Published Specifiction

4.4.3.2.1 Extending the XML Schema

Assuming that the simple model described above has been used, it is recommended that if a vendor wishes to extend an existing complexType they go through the following steps.

1. Create a new schema
2. Assign a unique specific namespace. There are no specific guidelines for what namespace is used as long as it is ensured to be globally unique.
3. Create elements for all of the extra content that will go into the messages. Add annotations to each new element and describe the purpose of the element and where it will be used.
4. If the vendor would like to make the extension public, the vendor can upload the specification at the HTNG extensions area at http://www.htng.org/tobedetermined.

Example
Let's assume a vendor would like to extend the (partial) schema described in the section above. The vendor would create the following schema to describe the additional elements needed

```
(01) <xs:schema targetNamespace="urn:some:unique:namespace"
      xmlns:tns="urn:some:unique:namespace"
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      elementFormDefault="qualified" attributeFormDefault="unqualified">
(02)    <xs:element name="MiddleName">
(03)       <xs:annotation>
(04)          <xs:documentation>This element extends the "Customer" element defined in the XYZ specification version
X.Y.</xs:documentation>
(05)       </xs:annotation>
(06)    </xs:element>
(07)    <xs:element name="Country">
(08)       <xs:annotation>
(09)          <xs:documentation>This element extends the "AddressType" complexType defined in the XYZ specification version
X.Y.</xs:documentation>
(10)       </xs:annotation>
(11)    </xs:element>
(12) </xs:schema>
```

The extended "Customer" element could then look like this:

```
(01) <Customer
    xmlns="urn:some:unique:namespace"
    xmlns:a="urn:some:unique:namespace">
```

```
(02)    <LastName>Smith</LastName>
(03)    <FirstName>John</FirstName>
(04)    <Address>
(05)      <Street>123 Main Street</Street>
(06)      <ZipCode>11111</ZipCode>
(07)      <a:Country>USA</a:Country>
(08)    </Address>
(09)    <a:MiddleName>Gary</a:MiddleName>
(10) </Customer>
```

4.4.3.2.2 Extending the WSDL

If a vendor needs to add functionality, it is highly recommended that a new WSDL is created with a new and unique namespace that will hold the new portType that will only contain the new/changed operations. The new WSDL may reference the WSDL and/or XML Schema that are part of the specification.

4.4.4 Suggested Workgroup Acronymns

Workgroup	Acronym
Architecture	ARC
Distribution Content Management	DCM
In-Room Technology	IRT
Payments	PMT
Property Distribution	PDS
Property Web Services	PWS

4.4.5 Guidelines for composing WSDL and XML Schemas

4.4.5.1 References to other WSDLs/Schemas

It is recommended to refrain from referencing WSDLs and/or XML Schemas that are part of other specifications. In the case that this is necessary, it is recommended to explicitly state the dependency on other standards in the specification text.

4.4.5.1.1 WSDL Structure

It is recommended that the following WS-I guidelines (http://www.ws-i.org) are observed

4.4.5.1.2 XML Schema

We recommend the following normanclature approaches:-

1. Element names should use Pascal casing, i.e. capitalize the first letter in a word.
2. Attribute names should use Camel casing, i.e. capitalize the first letter in a word except the first letter.
3. Elements be given names that can be easily understood

4.5 Security (if in use)

We recommend the use of WS Security as part of the normal development process. Below is a sample Synchronous message that uses the UsernameToken element as per WS-Security 1.0. WS-Addressing headers are also included. Note that the message is only using WS-Security for authentication, therefore no in-message signature or encryption is performed. The password is provided in clear text, and implies that the communication is secured at the transport level.

4.5.1 Security XML Example

```
(01) <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
   xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
   xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
(02)     <soap:Header>
(03)         <wsa:Action>http://htng.org/PWSWG/2006/05/BanquetEventOrder#BeoRequest</wsa:Action>
(04)         <wsa:MessageID>urn:uuid:29f43cdc-a621-4e2c-80af-3653545d5502</wsa:MessageID>
(05)         <wsa:ReplyTo>
(06)             <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
(07)         </wsa:ReplyTo>
(08)         <wsa:To>http://127.0.0.1/NIIS/BeoService/BeoService.asmx</wsa:To>
(09)         <wsse:Security soap:mustUnderstand="1">
(10)             <wsu:Timestamp wsu:Id="Timestamp-9f540437-93c5-4b9d-9d57-afad42eb007b">
(11)                 <wsu:Created>2006-10-30T16:07:46Z</wsu:Created>
(12)                 <wsu:Expires>2006-10-30T16:12:46Z</wsu:Expires>
(13)             </wsu:Timestamp>
(14)             <wsse:UsernameToken
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      wsu:Id="SecurityToken-9d1092c8-afdd-421c-9d78-044f6c25d777">
(15)                 <wsse:Username>TestUserName</wsse:Username>
(16)                 <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">TestPasswordValue</wsse:Password>
(17)                 <wsse:Nonce>+c1BfJJI3Slsm6J4l9Wk2w==</wsse:Nonce>
(18)                 <wsu:Created>2006-10-30T16:07:46Z</wsu:Created>
(19)             </wsse:UsernameToken>
(20)         </wsse:Security>
(21)     </soap:Header>
(22)     <soap:Body>
(23)     </soap:Body>
(24) </soap:Envelope>
```

4.6 Routing

Limited discussion has taken place to date on routing. The discussions to date have determined:-

1. If routing (or any type of intermediary forwarding mechanism) is used, a next-hop approach is highly recommended, i.e. every entity in the chain knows only of the next hop in the chain.

2. Every end point should be treated in the same way, i.e. there is no discernible difference between an intermediary (e.g. a router) and the logical final recipient.

4.7 Reliability

Discussion has taken place with respect to the applicability of WS-Reliability within Framework 2.1. The group determined that specific individual vendors may wish to agree and implement WS-Reliability within a particular workgroup or against an agreed set of messages this should be able to take place within the existing Framework 2.1 Header structure. Unless an agreement is specifically in place WS-Reliability is not a requirement for either Synchronous or Asynchronous processing.

A workgroup may include other SOAP headers within a Framework 2.1 message in order to implement other functions as required by their own implementation of an interface.

4.8 Handling Attachments

The workgroup recommends the use of MTOM as the methodology for handling attachments with messaging.

Chapter 5 The Asynchronous Communication Process

Asynchronous behavior is accomplished by implementing three “normal” web service methods, one on the service side that gets called by the consumer to start the asynchronous process (the method should have the _SubmitRequest suffix), and two methods that are implemented by the consumer to receive the result (suffixed _SubmitResult) or a fault (_SubmitFault).

5.1 ASYNC1

The response/reply message in each message exchange (the _SubmitRequest message exchange and _SubmitResult/_SubmitFault message exchange) are treated like normal and are correlated using WS-Addressing. In the sample, a UUID was used for CorrelationID :

```
(01) <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
      xmlns:htng="urn:tobedetermined">
(02)     <soap:Header>
(03)         <wsa:MessageID>uuid:214A50B2-E62E-4f8b-BD97-62ABE31E15C2</wsa:MessageID>
(04)         <wsa:ReplyTo>
(05)             <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
(06)         </wsa:ReplyTo>
(07)         <wsa:To>http://htng.org/abcspec</wsa:To>
(08)         <wsa:Action>http://htng.org/abcspec/XYZ_SubmitRequest</wsa:Action>
(09)         <htng:CorrelationID>uuid:2D11751F-916F-4c1f-B1FD-9D6D051AC90A</htng:CorrelationID>
(10)         <htng:ReplyTo>
(11)             <wsa:Address>http://abc:1234/MyClientAsync</wsa:Address>
(12)         </htng:ReplyTo>
(13)         <htng:FaultTo>
(14)             <wsa:Address>http://abc:1234/MyClientAsync</wsa:Address>
(15)         </htng:FaultTo>
(16)     </soap:Header>
(17)     <soap:Body>
(18)         <m:XYZ_SubmitRequest xmlns:m="http http://htng.org/abcspec">
(19)             ... The request ...
(20)         </m:XYZ_SubmitRequest>
(21)     </soap:Body>
(22) </soap:Envelope>
```

5.2 ASYNC2

The provider receives the message, initiates the asynchronous process and sends back an “empty” SOAP message

```
(01) <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" >
(02)     <soap:Header>
(03)         <wsa:MessageID>uuid:9971BF72-F42D-49ee-99DB-BEC28B6EDAF7</wsa:MessageID>
(04)         <wsa:RelatesTo>uuid:214A50B2-E62E-4f8b-BD97-62ABE31E15C2</wsa:RelatesTo>
(05)         <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
```

```
(06)         <wsa:Action>http://htng.org/abcspec/XYZ_SubmitRequestResponse</wsa:Action>
(07)     </soap:Header>
(08)     <soap:Body/>
(09) </soap:Envelope>
```

5.3 ASYNC3

The provider completes the asynchronous process, and invokes the _SubmitResult method on the consumer (the address was specified by the htng:ReplyTo element in the original _SubmitRequest method).

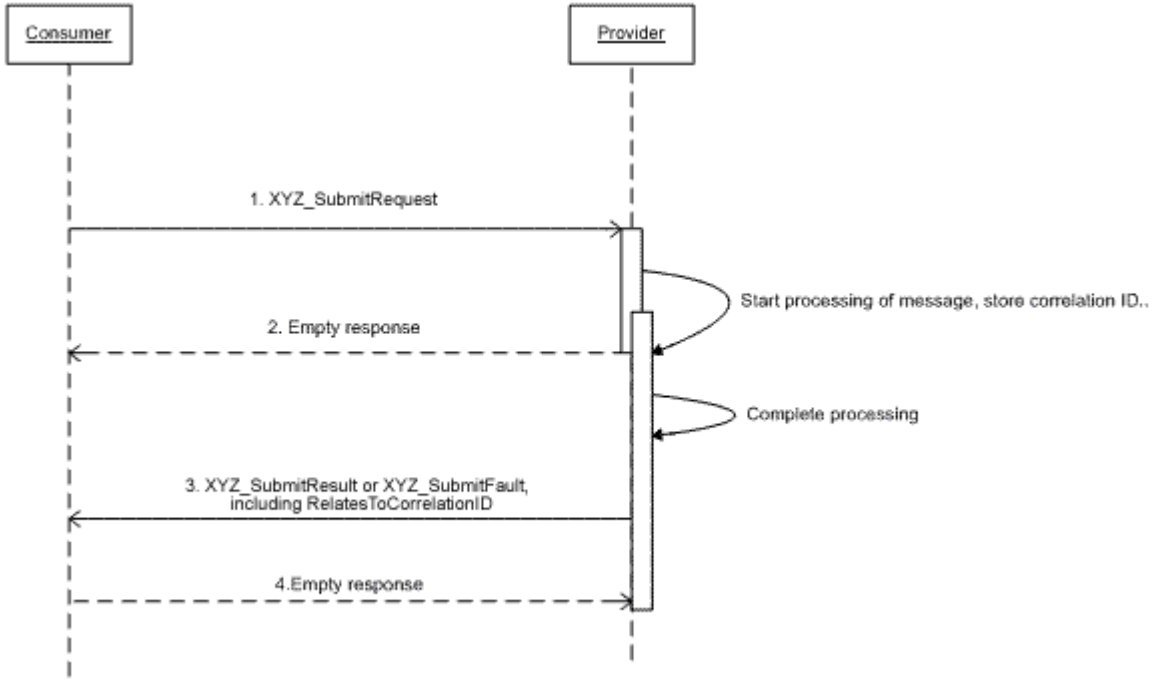
```
(01) <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
      xmlns:htng="urn:tobedetermined" >
(02)     <soap:Header>
(03)         <wsa:MessageID>uuid:C15EE2B2-B41C-44c4-901E-1032159CCC6A</wsa:MessageID>
(04)         <wsa:ReplyTo>
(05)             <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
(06)         </wsa:ReplyTo>
(07)         <wsa:To>http://htng.org/abcspec</wsa:To>
(08)         <wsa:Action> http://htng.org/abcspec/XYZ_SubmitResult</wsa:Action>
(09)         <htng:RelatesToCorrelationID>uuid:2D11751F-916F-4c1f-B1FD-9D6D051AC90A</htng:RelatesToCorrelationID>
(10)     </soap:Header>
(11)     <soap:Body>
(12)         <m:XYZ_SubmitResult xmlns:m="http://htng.org/abcspec">
(13)             ... The result ...
(14)         </m:XYZ_SubmitResult>
(15)     </soap:Body>
(16) </soap:Envelope>
```

5.4 ASYNC4

The consumer sends back an “empty” SOAP response message

```
(01) <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
(02)     <soap:Header>
(03)         <wsa:MessageID>uuid:6F400F52-8912-4dab-BEEB-FEDEC356979F</wsa:MessageID>
(04)         <wsa:RelatesTo>uuid:C15EE2B2-B41C-44c4-901E-1032159CCC6A</wsa:RelatesTo>
(05)         <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
(06)         <wsa:Action>http://xyz/MyService/XYZ_SubmitResultResponse</wsa:Action>
(07)     </soap:Header>
(08)     <soap:Body/>
(09) </soap:Envelope>
```

Diagrammatically this can be represented via the following sequence diagram:



5.5 Asynchronous Fault handling

There are four places where a fault can be reported:

1. In the response to _SubmitRequest method, i.e. the provider throws a fault back to the consumer.
   This fault will be sent back instead of the “empty” message.
   If this happens, the consumer can assume that NO asynchronous process was started and there will thus be no _SubmitResult or _SubmitFault call.

2. In the _SubmitFault method, i.e. the provider signals the consumer that the asynchronous operation completed as the result of a fault.

3. In the response to the _SubmitResult method, i.e. the consumer returns a fault back to the provider as a response to the _SubmitResult method.

4. In the response to the _SubmitFault method, i.e. the consumer returns a fault back to the provider as a response to the _SubmitFault method.

5.6 Asynchronous WSDL Structure

Create two port types, one that will hold the methods implemented by the providers and one implemented by the consumer to receive completion and error callbacks. For each _SubmitRequest method in the provider, create two callback methods, _SubmitResult and _SubmitFault, on the consumer's side.

5.6.1 Asynchronous WSDL Example

```
(01) <!--"Normal" port type that the provider implements.
   Contains both synchronous methods and the initiating method of asynchronous methods-->
(02) <portType name="ReservationProviderPortType">
(03)     <!--This is the normal synchronous version-->
(04)     <operation name="CreateReservation">
(05)         <input/>
(06)         <output/>
(07)         <fault/>
(08)     </operation>
(09)     <!--This is the asynchronous version-->
(10)     <operation name="CreateReservation_SubmitRequest">
(11)         <input message="tns:CreateReservation_SubmitRequestInputMessage"/>
(12)         <output message="tns:EmptyMessage"/>
(13)     </operation>
(14) </portType>
(15) <!--"Callback" port type that the caller implements to be able to receive completion
   callbacks (successes and failures) on asynchronous methods-->
(16) <portType name="ReservationAsyncCompletionPortType">
(17)     <operation name="CreateReservation_SubmitResult">
(18)         <input message="tns:CreateReservation_SubmitResultInputMessage"/>
(19)         <output message="tns:EmptyMessage"/>
(20)     </operation>
(21)     <operation name="CreateReservation_SubmitFault">
(22)         <input message="tns:CreateReservation_SubmitFaultInputMessage"/>
(23)         <output message="tns:EmptyMessage"/>
(24)     </operation>
(25) </portType>
```

Chapter 6 Simple HTNG Framework 2.1 Asynchronous Sample

A sample application is contained in the file *HTNG_Framework_2.0_Simple_Async_Sample.zip* in the *HTNG_Framework_2.0_Samples.zip* archive which can be downloaded from the same location as this specification.

The sample illustrates an implementation of the HTNG Framework 2.1 asynchronous pattern in WSE 3.0. In this sample, the notion of a “provider” and a “consumer” is used. The provider represents the “service” that is providing some business functionality, in this particular sample the business functionality consists of a dummy method called “CreateReservation”. The consumer represents the client application that is calling the provider to create a reservation.

The asynchronous pattern in the HTNG Framework 2.1 specifies an asynchronous operation as two separate message exchanges:

1. The consumer makes a regular synchronous http Web Service call (where the reply is received on the same connection as the request was sent on) to initiate the asynchronous request. An empty reply is sent back to indicate successful receipt of the request. The provider will kick off the asynchronous process.

2. The provider will make a regular synchronous http Web Service call (where the reply is received on the same connection as the request was sent on) to the consumer when the asynchronous process is completed to deliver the successful result or the unsuccessful fault of the asynchronous process.

The web methods that the provider and the consumer implements are described in separate portTypes in the WSDL to clearly separate the difference in the two roles that the consumer and provider plays in the message exchange.

This sample would have been fairly straightforward if WSE3 had out-of-the-box provided a way to accept http requests to a non-ASP.NET hosted process (by simply adding an “http” URI to the WSE “SoapReceivers” in the same way that it is possible to add a “soap.tcp” SoapReceiver, for example). While this is supported in WCF/Indigo, WSE3 does not support it, so this sample also contains a small Class Library “HttpSysTransport” originally written by Aaron Skonnard (with some minor enhancements) to make this possible in WSE3.

Appendix A SOAP Fault handling

It is recommended to use the standard SOAP fault model as described in the SOAP 1.1 and 1.2. This specification divides faults into two categories:

Faults that are known ahead of time, which are typically business-specific faults and are specific to each Web Service. These faults are well defined in the WSDL and XML Schema of the Web Service.

Faults that are not known ahead of time, which are typically implementation specific faults, e.g. communication faults, etc.

A.1. Declarative Approach

Faults that are known ahead of time and are business-specific should be declared in the WSDL to let the consumer of the Web Service know what types of business-related faults can be expected by calling a specific method. Each fault must also have a corresponding XML Schema element declared that uniquely describes the fault. Consider the following WSDL definition of a Web Service Method where there are two faults declared in the WSDL:

```
(01) <operation name="CreateReservation">
(02)     <input message="tns:CreateReservationInputMessage"/>
(03)     <output message="tns:CreateReservationOutputMessage"/>
(04)     <fault name="NoAvailability" message="tns:NoAvailabilityFaultMessage">
(05)         <documentation>Thrown if there is no longer any availability;</documentation>
(06)     </fault>
(07)     <fault name="InvalidData" message="tns:InvalidDataFaultMessage">
(08)         <documentation>Thrown if one or more fields in the data are not
                            filled in or did not validate.
                The contents of the fault will contain more
                            information
             </documentation>
(09)     </fault>
(10) </operation>
```

The message definition for the NoAvailabiltyFaultMessage and InvalidDataFaultMessage look like this:

```
(01) <message name="NoAvailabilityFaultMessage">
(02)     <part name="parameters" element="tns:NoAvailabilityFault"/>
(03) </message>
(04) <message name="InvalidDataFaultMessage">
(05)     <part name="parameters" element="tns:InvalidDataFault"/>
(06) </message>
```

And the XML Schema elements NoAvailabilityFault and InvalidDataFault look like this:

```
(01) <xs:element name="NoAvailabilityFault">
(02)     <xs:complexType>
(03)         <xs:sequence>
(04)             <xs:element name="FirstAvailableDateAndTime" type="xs:dateTime"/>
(05)         </xs:sequence>
```

```
(06)     </xs:complexType>
(07) </xs:element>
(08) <xs:element name="InvalidDataFault">
(09)     <xs:complexType>
(10)         <xs:sequence>
(11)             <xs:element name="Field" maxOccurs="unbounded">
(12)                 <xs:complexType>
(13)                     <xs:sequence>
(14)                         <xs:element name="Reason" type="xs:string"/>
(15)                     </xs:sequence>
(16)                     <xs:attribute name="name" type="xs:string" use="required"/>
(17)                 </xs:complexType>
(18)             </xs:element>
(19)         </xs:sequence>
(20)     </xs:complexType>
(21) </xs:element>
```

A SOAP 1.2 envelope containing the SOAP fault for InvalidData would look like this:

```
(01) <env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
         xmlns:t="http://new.webservice.namespace"
         xmlns:xml="http://www.w3.org/XML/1998/namespace">
   … WS-Addressing, Security, etc headers omitted …
(02)     <env:Body>
(03)         <env:Fault>
(04)             <env:Code>
(05)                 <env:Value>env:Sender</env:Value>
(06)                 <env:Subcode>
(07)                     <env:Value>t:InvalidData</env:Value>
(08)                 </env:Subcode>
(09)             </env:Code>
(10)             <env:Reason>
(11)                 <env:Text xml:lang="en">Some of the fields were
                                         invalid</env:Text>
(12)             </env:Reason>
(13)             <env:Detail>
(14)                 <t:InvalidDataFault>
(15)                     <t:Field name="FirstName">
(16)                         <t:Reason>A first name must set</t:Reason>
(17)                     </t:Field>
(18)                     <t:Field name="DOB">
(19)                         <t:Reason>The date of birth cannot be in the
                                       future</t:Reason>
(20)                     </t:Field>
(21)                 </t:InvalidDataFault>
(22)             </env:Detail>
(23)         </env:Fault>
```

```
(24)     </env:Body>
(25) </env:Envelope>
```

And the SOAP 1.2 envelope for the NoAvailability fault would look like this:

```
(01) <env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
         xmlns:t="http://new.webservice.namespace"
         xmlns:xml="http://www.w3.org/XML/1998/namespace">
   … WS-Addressing, Security, etc headers omitted …
(02)     <env:Body>
(03)         <env:Fault>
(04)             <env:Code>
(05)                 <env:Value>env:Sender</env:Value>
(06)                 <env:Subcode>
(07)                     <env:Value>t:NoAvailability</env:Value>
(08)                 </env:Subcode>
(09)             </env:Code>
(10)             <env:Reason>
(11)                 <env:Text xml:lang="en">There was no availability
                                             left to complete the reservation.
                      </env:Text>
(12)             </env:Reason>
(13)             <env:Detail>
(14)                 <t:NoAvailabilityFault>
(15)                     <t:FirstAvailableDateAndTime>2006-12-17T09:30:00Z</t:FirstAvailableDateAndTime>
(16)                 </t:NoAvailabilityFault>
(17)             </env:Detail>
(18)         </env:Fault>
(19)     </env:Body>
(20) </env:Envelope>
```

Please note that the example above represents the SOAP message carrying a fault as it might look in a synchronous session, or in an asynchronous session if the fault were to occur in the context of the call.
Fault(s) that occur in the asynchronous process initiated by an asynchronous call would be submitted to the consumer via the _SubmitFault operation, therefore the SOAP message might look close to the following:

```
(01) <env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
         xmlns:t="http://new.webservice.namespace"
         xmlns:xml="http://www.w3.org/XML/1998/namespace">
   … WS-Addressing, Security, RelatesToCorrelationID etc. headers omitted …
(02)   <env:Body>
(03)     <m:XYZ_SubmitFault>
(04)       <env:Fault>
(05)         <env:Code>
(06)           <env:Value>env:Sender</env:Value>
(07)             <env:Subcode>
(08)               <env:Value>t:NoAvailability</env:Value>
```

```
(09)            </env:Subcode>
(10)          </env:Code>
(11)        <env:Reason>
(12)          <env:Text xml:lang="en">There was no availability left to complete the reservation.</env:Text>
(13)        </env:Reason>
(14)        <env:Detail>
(15)          <t:NoAvailabilityFault>
(16)            <t:FirstAvailableDateAndTime>2006-12-17T09:30:00Z</t:FirstAvailableDateAndTime>
(17)          </t:NoAvailabilityFault>
(18)        </env:Detail>
(19)      </env:Fault>
(20)    </m:XYZ_SubmitFault>
(21)  </env:Body>
(22) </env:Envelope>
```

Appendix B SOAP Exception Handling

A sample application has been created that provides an example of SOAP exception handling.
This example is contained in the file *HTNG_Framework_2.0_SOAP_Fault_Handling_with_WSE_Sample.zip* in the *HTNG_Framework_2.0_Samples.zip* archive which can be downloaded from the same location as this specification.

The package contains examples and utilities for how to more easily use structured SOAP exceptions in the Microsoft WSE environment. Microsoft WSE does not fully implement structured SOAP exceptions (those exceptions that are identified by the contents of the <Detail> element in the SOAP fault element).

The method described in the example is aimed to be similar to what is seen in other web service toolkits , e.g. AXIS, and Microsoft's upcoming communication framework "WCF" (previously known as "Indigo"). Once WCF is released, porting structured SOAP exception handling using this method will be straightforward. In order to make this work, this method consists of two parts:

1. A utility to generate the exception classes that correspond to a WSDL <fault> element that refers to a message with an XML Schema element describing the content. This utility is provided in this package as a console application with full source called "GenerateFaultWrappersFromWsdl." The exception classes that are created all inherit from SoapException, so it is easy and straightforward to use them on the service side, simply "throw" the strongly typed exception and you are done. A little bit more work is required on the client side to translate the SoapException to a strongly typed exception, for this purpose the GenerateFaultWrappersFromWsdl utility also generates a class to map a SoapException to a strongly typed exception.

2. In order to make the translation of the exception on the client side to happen seamlessly to the caller of the web service proxy, the proxy class that gets generated by the Wsdl/WseWsdl3 utilities needs to be extended. With .NET2 this can be done in a non-invasive way since the proxy web service class is generated with the "partial" attribute. Please see the MathServiceExtension.cs source file in "Contract.Client".

NOTE: One alternative to this could also be to implement a custom Policy Assertion in WSE3 and throw the strongly typed exception from there. Unfortunately, WSE3 will wrap all non-SoapException exceptions thrown from a policy assertion, which would not let the caller use the plain try {} catch {} pattern.

This will allow service implementers use the following style of code :

```
throw new MyStronglyTypedException(…);
```

And client implementers use the following style of code :

```
try
{
    Mywebserviceproxy.SomeCall(…);
}
catch(MyStronglyTypedException myException)
{
    // The content of the exception is in the myException.TypedDetail property
}
```

Appendix C Math(A+B) sample project

A sample application is provided that implements a simple A+B application. This application should be used as a minimum primer to become familiar with Framework 2.1.

This example is contained in the file *HTNG_Framework_2.0_Math_A+B_Sample.zip* in the *HTNG_Framework_2.0_Samples.zip* archive which can be downloaded from the same location as this specification.

This example contains

- Custom UserTokenManager that demonstrates and discusses how to implement proprietary authentication.
- Custom Policy implemented in code to require SOAP action header and UsernameToken
- Web service implementation of add, subtract, multiply, divide.
- Console application client that demonstrates how to call the service with a username token, etc.

Appendix D Payment Posting Example

A sample application is provided that implements the payment posting message. This message has been most recently used by the Single Guest Itinerary workgroup.

The sample is contained in the file *HTNG_Framework_2.0_Payment_Posting_Sample.zip* in the *HTNG_Framework_2.0_Samples.zip* archive which can be downloaded from the same location as this specification.

The example implements a Web Service at http://htng.org/PWSWG/2006/04/SingleGuestItinerary#PostPayment and was developed using a small part of Activity.wsdl designed by Guest Itinerary Workgroup. The WSDL that was used can be found in the Schema subfolder. Please note, no custom Fault conditions were defined in the WSDL.

The following pre-requisites are required on the system in order to build and run the sample:

- MS IIS
- MS .NET Framework v2.0
- MS Visual Studio 2005
- MS Web Service Enhancements v3.0

Appendix E Additional Examples

Additional examples have been provided by the Workgroup to assist in people's understanding of Framework 2.1.

1. HTNG 2.0 OTA Ping Service

   This example is contained in the file *HTNG_Framework_2.0_OTA_Ping_Sample.zip* in the *HTNG_Framework_2.0_Samples.zip* archive which can be downloaded from the same location as this specification.

   The zip archive contains:

   • a WSDL which describes the service and should allow the generation of the client code
   • an example for the request and response including the HTTP header To use the service, use:

   username = trust
   password = xyz

   The service is programmed in Java and uses AXIS 1.4.

   For addressing, http://schemas.xmlsoap.org/ws/2008/08 was used.

2. WS-Security example

   This example is contained in the file *HTNG_Framework_2.0_WS_Security_Sample.xml* in the *HTNG_Framework_2.0_Samples.zip* archive which can be downloaded from the same location as this specification.

   This is a sample Synchronous message that uses the UsernameToken element as per WS-Security 1.0. WS-Addressing headers are also included. Please note that the message is only using WS-Securityfor authentication, therefore no in-message signature or encryption is performed. The password is provided in clear text, and implies that the communication is secured at the transport level.

3. HTNG 2.0 Asynchronous OTA Ping Sample for .NET 2.0 and WSE 3.0
   This is a simple example of an OTA ping using the Asynchronous processing model 2

Part 2 ***Single Guest Itinerary Version 1.0.1***

Chapter 7 Contributors

7.1 Message Specification

Company / Organization	Contributor
FCS Computer Systems	Chris Armour
Four Seasons Hotels and Resorts	Natasha Nelson
Hotel Technology Next Generation	Douglas Rice
MICROS Systems, Inc.	Tom Gresham Victor Konopik
OpenCourse Solutions	Kristofer Agren
PAR Springer-Miller Systems, Inc.	Sophie Grigg Alex Lobakov Penka Sevova
TAC Informationtechnologie GmbH	Bernhard Rappold Thomas Rossler

7.2 Messafe Flow Documentation

Company / Organization	Contributor
Four Seasons Hotels and Resorts	Natasha Nelson
MICROS Systems, Inc.	Tom Gresham Victor Konopik
OpenCourse Solutions	Kristofer Agren
PAR Springer-Miller Systems, Inc.	Sophie Grigg Alex Lobakov Penka Sevova
TAC Informationtechnologie GmbH	Bernhard Rappold Thomas Rossler

NOTE: Version 1.0.1, posted on 27 May 2010, includes an addendum to Single Guest Itinerary, which defines the roles and use cases for the specification, and which was inadvertently omitted from the original document. The addendum is posted as an additional document on the Open Group site, where Property Web Services is posted.

Chapter 8 Activity Service

The WSDL for the HTNG Activity Reservation System defines the "ActivityService" web service. This web service consists of 17 functions defined in six port types. The port types are:

- **ActivityPortType**
  - Generic activity functions which are implemented by both the Itinerary Consolidation System and the Activity Reservation System.
- **ActivityProviderBinding**
  - Activity functions only applicable to the Activity Reservation System
- **ActivitySyncBinding**
  - Activity functions only applicable to the Itinerary Consolidation System
- **NameBinding**
  - Support functions for guest profile information, applicable to both the Itinerary Consolidation System and the Activity Reservation System.
- **ReservationProviderBinding**
  - Support functions for reservation information, applicable to the Activity Reservation System.
- **ReservationSyncBinding**
  - Support functions for reservation information, applicable to the Itinerary Consolidation System.

8.1 Supporting Schemas

The Activity Web Service imports five schemas which define the data payloads required by the functional messages. These are:

- **Activity**.xsd
  - Defines messages specific to activities.
- **Common**.xsd
  - Defines generic types used by various other elements.
- **Name**.xsd
  - Defines data types specific to a guest profile.
- **PmtPosting**.xsd
  - Defines data required for payment posting.
- **Reservation**.xsd
  - Defines data applicable to a guest reservation.

Common elements used in several messages are included in the Common Schema Elements section of this document.

8.2 Soap Header

All messages must include a soap header that conforms to the WSAddressing and WSSecurity specifications. Receiving systems may optionally require that the wsa:To element match a specific destination address, and therefore, this element should be configurable at run time. In addition, all sending systems must identify themselves by specifying a wsa:From element unique for their system. Typically, this is in the form of "URN:<*system*>". The wsa:ReplyTo address may be the anonymous form described in the August, 2004 specification.

WSSecurity may be optionally implemented. Providers are only required to support basic plain text authentication with a username and password. The user credentials will be limited to a single login per system, agreed between two vendors. There is no requirement to support multiple user logins from a single vendor through this interface.

8.2.1 Sample Header Message

```
<soap:Header>
   <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#PostPayment</wsa:Action>
   <wsa:From>
      <wsa:Address>urn:SPASOFT</wsa:Address>
   </wsa:From>
   <wsa:MessageID>urn:uuid:e63d962e-94b6-434c-89ea-1c5ae2e0f8ba</wsa:MessageID>
   <wsa:ReplyTo>
      <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
   </wsa:ReplyTo>
   <wsa:To>http://www.micros.com/HTNGActivity/</wsa:To>
   <wsse:Security soap:mustUnderstand="1">
      <wsu:Timestamp wsu:Id="Timestamp-015e2941-114a-466e-87b6-8c431b9f5c44">
         <wsu:Created>2006-10-26T12:49:17Z</wsu:Created>
         <wsu:Expires>2006-10-26T12:54:17Z</wsu:Expires>
      </wsu:Timestamp>
      <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
         wsu:Id="SecurityToken-627f1ab1-338a-451f-9829-84f248e57ad8">
         <wsse:Username>HTNG</wsse:Username>
         <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">HTNG</wsse:Password>
         <wsse:Nonce>qls5nr9rM7VaUYAwhrHSoA==</wsse:Nonce>
         <wsu:Created>2006-10-26T12:49:17Z</wsu:Created>
      </wsse:UsernameToken>
   </wsse:Security>
</soap:Header>
```

Chapter 9 Activity Port Type

The Activity Port defines four functions which should be implemented by both the Itinerary Consolidation System and the Activity Reservation System. These are

- ActivityLookup
  - Function to request a list of guest records with or without activities matching one or more search criteria.
- CreateActivity
  - Function to create an activity record.
- UpdateActivity
  - Function to update an activity record.
- CancelActivity
  - Function to cancel a scheduled activity.

9.1 Activity Lookup

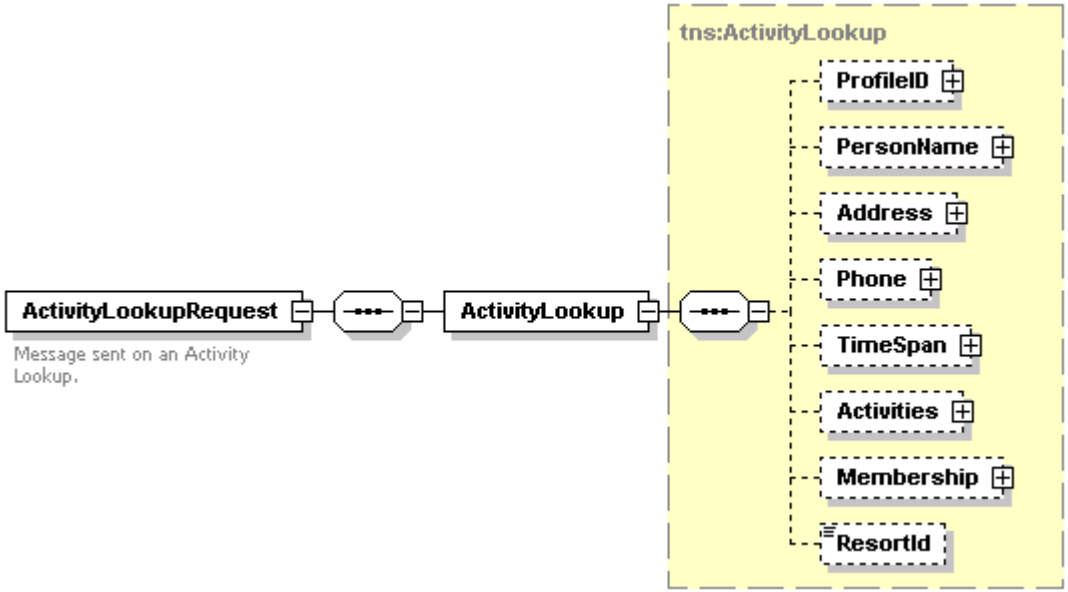
An activity lookup request message is submitted in order to return a list of activities that match one or more criteria. The items which may be submitted in the request are various aspects of the guest profile record such as a guest's name, guest ID, telephone number, or membership number. The requesting system submits an ActivityLookupRequest message, and the responding system returns and ActivityLookupResponse. This message is applicable to both the Activity Reservation System and the Itinerary Consolidation System.

It is required at a minimum that the requesting system send either a guest profile identifier valid in the external system, or a guest's last name. The receiving system may use the remainder of the filter information in order to further limit the search. However, as the request message includes a substantial number of data elements (due to message reuse), the receiving system is not required to process every filter item sent. The responding system should include profiles matching the search criteria regardless whether there are activities attached to the profile or not.

As this operation is most often performed through user interaction, it is important that the lookup respond in a reasonable time frame. If the criteria are too broad, the responding system should limit the result set to a reasonable number of items. This reduces the time needed to collect and respond with the data, and reduces the length of the data returned. As this operation may often return a list of possible selections, the number should also be limited since an operator would not typically want to scroll through more than a few dozen records.

Port	ActivityPortType
Binding	ActivityBinding
Operation	ActivityLookup
Soap Action	http://htng.org/PWSWG/2006/08/SingleGuestItinerary#ActivityLookup
Input	ActivityLookupRequest
Output	ActivityLookupResponse
Primary Schema	Activity.xsd
Role(s) Implemented	Itinerary Consolidation System, Activity Reservation System

9.1.1 ActivityLookupRequest



ActivityLookupRequest

```
<xs:element name="ActivityLookupRequest">
    <xs:annotation>
    <xs:documentation>Message sent on an Activity Lookup.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ActivityLookup" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types" type="q1:ActivityLookup"
                />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
ActivityLookup	element	ActivityLookup	required	Contains search criteria used for activity lookup.

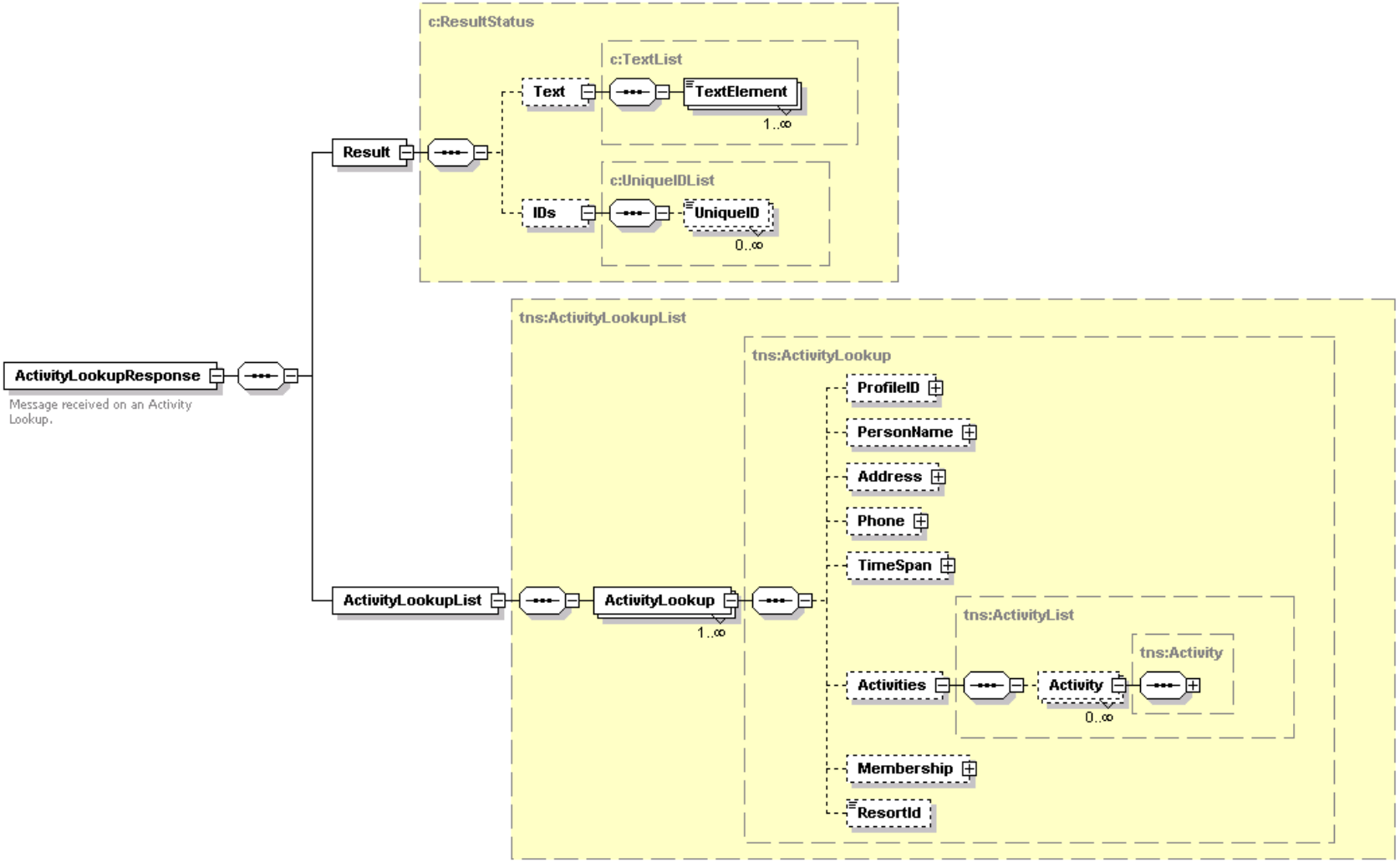
ActivityLookup

ActivityLookup

```
<xs:complexType name="ActivityLookup">
    <xs:annotation>
    <xs:documentation>Used in the request message to describe the "filter" when executing an activity lookup. Also used in the response message to return
        the activities matching the filter criteria.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element minOccurs="0" name="ProfileID" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q1:UniqueIDList" />
        <xs:element minOccurs="0" name="PersonName" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q2:PersonName" />
        <xs:element minOccurs="0" name="Address" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q3:Address"
            />
        <xs:element minOccurs="0" name="Phone" xmlns:q4="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q4:Phone" />
        <xs:element minOccurs="0" name="TimeSpan" xmlns:q5="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
            type="q5:OGTimeSpan" />
        <xs:element minOccurs="0" name="Activities" xmlns:q6="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
            type="q6:ActivityList" />
        <xs:element minOccurs="0" name="Membership" xmlns:q7="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q7:Membership" />
        <xs:element minOccurs="0" name="ResortId" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="externalSystemId" type="xs:string" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
externalSystemId	attribute	string	none	Not used in this context.
ProfileID	element	UniqueIDList	optional	Filter to lookup activities belonging to a specific guest profile record. The ID value applies to destination system (not requesting guest ID). Although an ID list is specified, the responding system is required to only process the first ID in the list (it may optionally process additional ID's). At a minimum, either a profile ID or a guest last name must be specified in this message.
PersonName	element	PersonName	optional	The name of the guest to search activities for. At a minimum, the guest's last name must be specified (unless searching by profile ID).
Address	element	Address	optional	The guest address details to further limit the search. The responding may optionally process this filter but is not required to do so.
Phone	element	Phone	optional	The guest telephone number to further limit the search. The responding may optionally process this filter but is not required to do so.
TimeSpan	element	OGTimeSpan	optional	A period of time for which to return activities for. If this optional criterion is applied, then any activity that begins, spans, or ends during this time period should be returned.
Activities	element	ActivityList	optional	An optional filter to search for specific activity matches.
Membership	element	Membership	optional	An optional filter to enhance guest name search by including membership information.
ResortId	element	string	optional	An optional filter when searching in a specific property. Typically, this is not applicable in this context.

9.1.2 ActivityLookupResponse



ActivityLookupResponse

```
<xs:element name="ActivityLookupResponse">
    <xs:annotation>
    <xs:documentation>Message received on an Activity Lookup.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Result" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:ResultStatus" />
            <xs:element name="ActivityLookupList" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
                type="q2:ActivityLookupList" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
Result	element	ResultStatus	required	Lookup result. If successful, the responding system is only required to set the resultStatusFlag as SUCCESS, and return the data in the ActivityLookupList.
ActivityLookupList	element	ActivityLookupList	required	A list of matching activity records.

ActivityLookupList

ActivityLookupList

```
<xs:complexType name="ActivityLookupList">
    <xs:annotation>
    <xs:documentation>A collection of Activity Lookup items.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element maxOccurs="unbounded" name="ActivityLookup" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
            type="q1:ActivityLookup" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
ActivityLookup	element	ActivityLookup	required / multiple	A container for one or more ActivyLookup records.

ActivityLookup

ActivityLookup

```
<xs:complexType name="ActivityLookup">
    <xs:annotation>
    <xs:documentation>Used in the request message to describe the "filter" when executing an activity lookup. Also used in the response message to return
        the activities matching the filter criteria.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element minOccurs="0" name="ProfileID" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q1:UniqueIDList" />
```

```
        <xs:element minOccurs="0" name="PersonName" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q2:PersonName" />
        <xs:element minOccurs="0" name="Address" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q3:Address"
            />
        <xs:element minOccurs="0" name="Phone" xmlns:q4="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q4:Phone" />
        <xs:element minOccurs="0" name="TimeSpan" xmlns:q5="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
            type="q5:OGTimeSpan" />
        <xs:element minOccurs="0" name="Activities" xmlns:q6="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
            type="q6:ActivityList" />
        <xs:element minOccurs="0" name="Membership" xmlns:q7="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q7:Membership" />
        <xs:element minOccurs="0" name="ResortId" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="externalSystemId" type="xs:string" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
externalSystemId	attribute	string	none	Not used in this context.
ProfileID	element	UniqueIDList	optional	The profile ID from the responding system for the activity record. This is operationally required in this context.
PersonName	element	PersonName	optional	The guest name responsible for the activity. This is required in this context.
Address	element	Address	optional	The guest's primary address.
Phone	element	Phone	optional	The guest's primary phone number.
TimeSpan	element	OGTimeSpan	optional	The time the activity starts and its duration. Required in this context.
Activities	element	ActivityList	optional	A list of activities scheduled for this guest.
Membership	element	Membership	optional	The guest membership information.
ResortId	element	string	optional	The property code if applicable.

ActivityList

ActivityList

```
<xs:complexType name="ActivityList">
    <xs:annotation>
    <xs:documentation xml:lang="en">A collection of Activity objects.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Activity">
            <xs:complexType>
                <xs:complexContent mixed="false">
                    <xs:extension xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types" base="q1:Activity" />
                </xs:complexContent>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Activity	element	Activity	optional / multiple	A collection of activity records.

9.1.3 Sample Message

[*Request*]

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#ActivityLookup</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:OPERA</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:73dbe98b-7b79-409c-bf6c-0cd70c62e12e</wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.springermiller.com/HTNG_2_0/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-5f1dfcd2-757c-4c97-a4ee-f932f2058ea5">
                <wsu:Created>2006-11-02T19:35:39Z</wsu:Created>
                <wsu:Expires>2006-11-02T19:40:39Z</wsu:Expires>
            </wsu:Timestamp>
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
ad2009a7-088f-41a0-accb-1ea14a7b50a0">
                <wsse:Username>OPERA</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">OPERA</wsse:Password>
                <wsse:Nonce>Z3EjAxKiTnO7p6X7TvMMPg==</wsse:Nonce>
                <wsu:Created>2006-11-02T19:35:39Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <ActivityLookupRequest xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types">
            <ActivityLookup externalSystemId="SPASOFT">
                <PersonName>
                    <FirstName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">VERA</FirstName>
                    <LastName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">LINDEN</LastName>
                </PersonName>
                <Address>
```

```
                <CityName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"/>
                <StateProv xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"/>
                <CountryCode xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"/>
                <PostalCode xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"/>
            </Address>
            <Phone>
                <PhoneNumber xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"/>
            </Phone>
            <Activities>
                <Activity>
                    <TimeSpan>
                        <Start>2006-11-02T00:00:00</Start>
                    </TimeSpan>
                </Activity>
            </Activities>
            <Membership>
                <MembershipNumber xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"/>
            </Membership>
            <ResortId>HTNG1</ResortId>
        </ActivityLookup>
    </ActivityLookupRequest>
  </soap:Body>
</soap:Envelope>
```


[*Response*]


```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
  <soap:Header>
    <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#ActivityLookupResponse</wsa:Action>
    <wsa:MessageID>urn:uuid:359ee919-8a92-4be1-a489-197eeedf8c2c</wsa:MessageID>
    <wsa:RelatesTo>urn:uuid:73dbe98b-7b79-409c-bf6c-0cd70c62e12e</wsa:RelatesTo>
    <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
    <wsse:Security>
      <wsu:Timestamp wsu:Id="Timestamp-c62c11cc-ea8d-47f0-a0f6-413034ec6f0d">
        <wsu:Created>2006-11-02T19:37:08Z</wsu:Created>
        <wsu:Expires>2006-11-02T19:42:08Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <ActivityLookupResponse xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">
```

```
<Result code="String" resultStatusFlag="SUCCESS">
    <c:Text>
        <c:TextElement language="en"/>
    </c:Text>
    <c:IDs>
        <c:UniqueID source="SPASOFT">603</c:UniqueID>
    </c:IDs>
</Result>
<ActivityLookupList>
    <ActivityLookup externalSystemId="OPERA">
        <ProfileID>
            <c:UniqueID source="SPASOFT">603</c:UniqueID>
        </ProfileID>
        <PersonName>
            <c:FirstName>Vera</c:FirstName>
            <c:LastName>Linden</c:LastName>
        </PersonName>
        <Address>
            <c:AddressLine/>
            <c:CityName>Mount Albert</c:CityName>
            <c:StateProv>ON</c:StateProv>
            <c:CountryCode>CA</c:CountryCode>
            <c:PostalCode>L0G 1M0</c:PostalCode>
        </Address>
        <Phone phoneRole="" phoneType="">
            <c:PhoneNumber>752-1800</c:PhoneNumber>
        </Phone>
        <TimeSpan>
            <Start>2006-11-02T10:15:00</Start>
            <End>2006-11-02T10:45:00</End>
        </TimeSpan>
        <Activities>
            <Activity status="BOOK">
                <ActivityIDs>
                    <c:UniqueID source="SPASOFT">3632</c:UniqueID>
                </ActivityIDs>
                <ActivityType>Spa</ActivityType>
                <Location>Tranquility</Location>
                <Name language="en">Tranquility</Name>
                <NumberOfPersons>1</NumberOfPersons>
                <TimeSpan>
                    <Start>2006-11-02T10:15:00</Start>
                    <End>2006-11-02T10:45:00</End>
                </TimeSpan>
                <Duration>
```

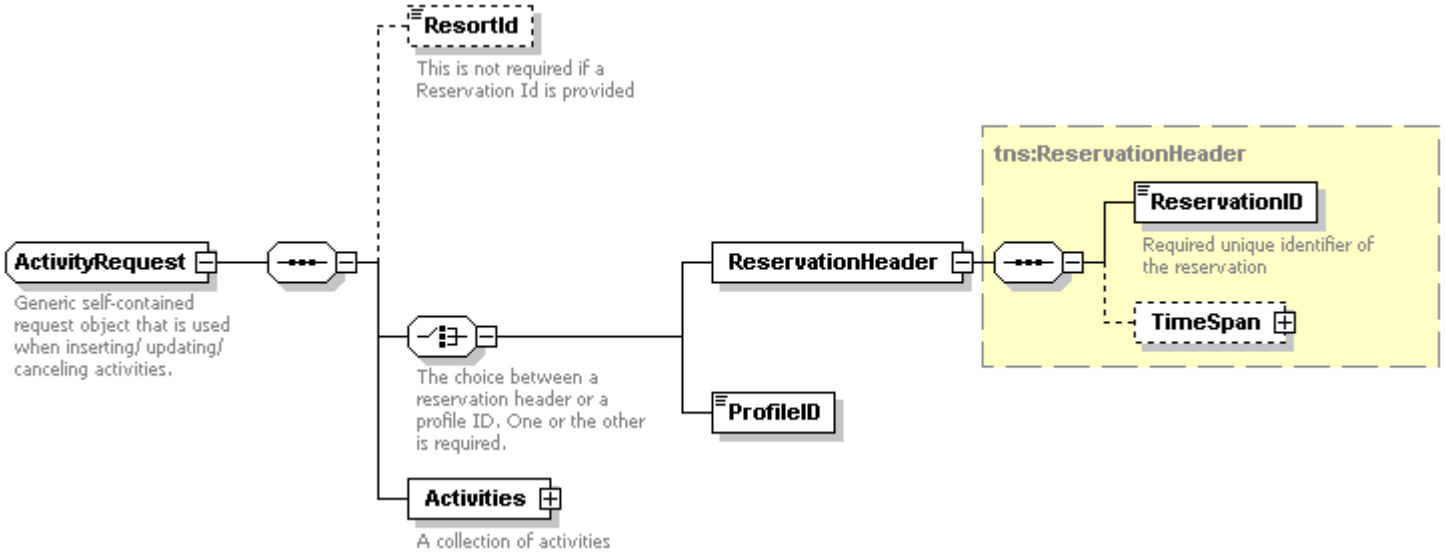
```
                <Start>2006-11-02T10:15:00</Start>
                <End>2006-11-02T10:45:00</End>
            </Duration>
            <Description language="en">Honey Bath</Description>
            <Amount currencyCode="US"/>
            <Extensions>
                <ActivitySpecific>Provider: Farrows Pierre</ActivitySpecific>
                <ActivitySpecific>Facility: Hydrotherapy Tub 1</ActivitySpecific>
                <ActivitySpecific>ReservationNo: R1184</ActivitySpecific>
            </Extensions>
        </Activity>
    </Activities>
  </ActivityLookup>
 </ActivityLookupList>
</ActivityLookupResponse>
</soap:Body>
</soap:Envelope>
```


9.2 Create Activity

The create activity message is used to add a new activity record. The requesting system submits an CreateActivityRequest (ActivityRequest) message, and the responding system returns and CreateActivityResponse (ActivityResponse). This message is primarily applicable to the Itinerary Consolidation System, but may also be implemented by the Activity Reservation System. In order to create an activity in an external system, the requesting system must send either the guest profile ID or the guest reservation number as defined in the receiving system. If successful, the response message should include a list of activity ID's created in the external system. Note that multiple activities may be created against a single reservation or guest profile with one request.

Port	ActivityPortType
Binding	ActivityBinding
Operation	CreateActivity
Soap Action	http://htng.org/PWSWG/2006/08/SingleGuestItinerary#CreateActivity
Input	CreateActivityRequest
Output	CreateActivityResponse
Primary Schema	Activity.xsd
Role(s) Implemented	Itinerary Consolidation System, Activity Reservation System

9.2.1 ActivityRequest



ActivityRequest

```
<xs:complexType name="ActivityRequest">
    <xs:annotation>
    <xs:documentation>Generic self-contained request object that is used when inserting/ updating/ canceling activities.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element minOccurs="0" name="ResortId" type="xs:string">
            <xs:annotation>
            <xs:documentation>This is not required if a Reservation Id is provided</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:choice>
            <xs:annotation>
            <xs:documentation xml:lang="en">The choice between a reservation header or a profile ID. One or the other is required.</xs:documentation>
            </xs:annotation>
            <xs:element name="ReservationHeader" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
                type="q1:ReservationHeader" />
            <xs:element name="ProfileID" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q2:UniqueID" />
        </xs:choice>
        <xs:element name="Activities" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types" type="q3:ActivityList">
            <xs:annotation>
```

```
            <xs:documentation>A collection of activities</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
ResortId	element	string	optional	The property code to create the activity for.
ReservationHeader	element	ReservationHeader	required	Identifies the guest reservation to create the activity against. Either this element or the ProfileID must be specified.
ProfileID	element	UniqueID	required	Identifies the guest profile to create the activity against. Either this element of the ReservationHeader must be specified.
Activities	element	ActivityList	required	A collection of activities to create for the reservation or guest specified.

ReservationHeader

ReservationHeader

```
<xs:complexType name="ReservationHeader">
    <xs:annotation>
    <xs:documentation>Used to uniquely identify a reservation using it's internal ID, it's begin date and duration or end date.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="ReservationID" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueID">
            <xs:annotation>
            <xs:documentation>Required unique identifier of the reservation</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="TimeSpan" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
            type="q2:OGTimeSpan" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
ReservationID	element	UniqueID	required	The reservation ID as defined by the receiving system.
TimeSpan	element	OGTimeSpan	optional	Not required in this context.

UniqueID

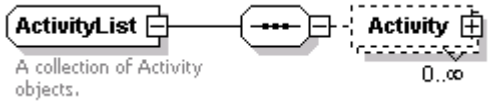
UniqueID

```
<xs:complexType name="UniqueID">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="source" type="xs:string" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments

UniqueID	extension	string	required	The guest profile ID as defined in the receiving system.
source	attribute	string	required	The source attribute that identifies the receiving system.

ActivityList

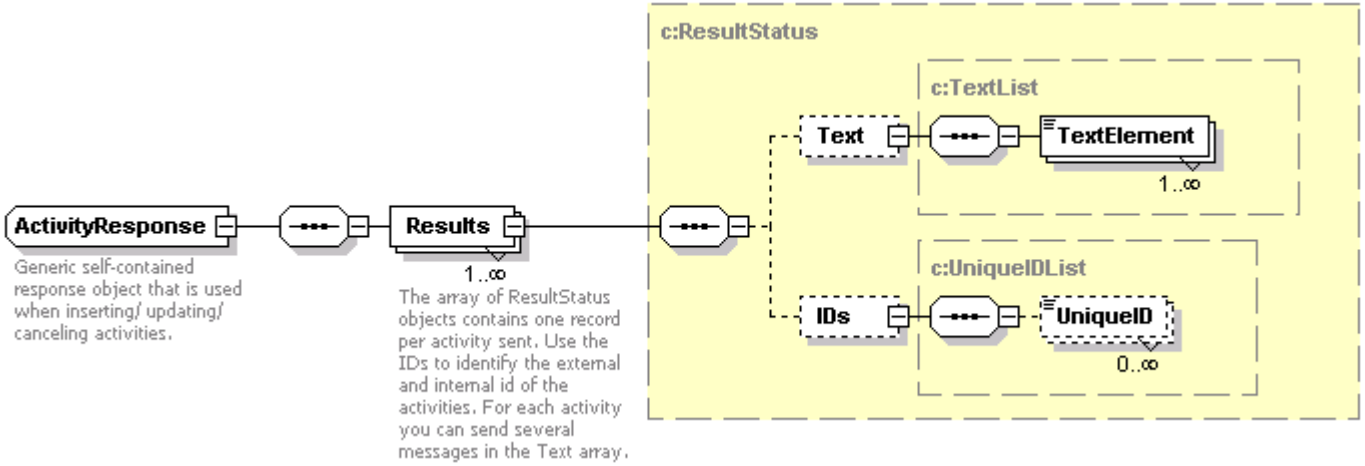


ActivityList

```
<xs:complexType name="ActivityList">
    <xs:annotation>
    <xs:documentation xml:lang="en">A collection of Activity objects.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Activity">
            <xs:complexType>
                <xs:complexContent mixed="false">
                    <xs:extension xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types" base="q1:Activity" />
                </xs:complexContent>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Activity	element	Activity	optional / multiple	A collection of activity records to create in the external system.

9.2.2 ActivityResponse



ActivityResponse

```
<xs:complexType name="ActivityResponse">
    <xs:annotation>
    <xs:documentation>Generic self-contained response object that is used when inserting/ updating/ canceling activities.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element maxOccurs="unbounded" name="Results" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q1:ResultStatus">
            <xs:annotation>
            <xs:documentation>The array of ResultStatus objects contains one record per activity sent. Use the IDs to identify the external and internal id of
                the activities. For each activity you can send several messages in the Text array.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Results	element	ResultStatus	required / multiple	A collection of result elements, one for each activity requested.

ResultStatus

ResultStatus

```
<xs:complexType name="ResultStatus">
    <xs:sequence>
```

```
        <xs:element minOccurs="0" name="Text" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:TextList" />
        <xs:element minOccurs="0" name="IDs" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q2:UniqueIDList"
            />
    </xs:sequence>
    <xs:attribute name="resultStatusFlag" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q3:ResultStatusFlag" />
    <xs:attribute name="code" type="xs:string" use="optional" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
resultStatusFlag	attribute	ResultStatusFlag	required	The result (*SUCCESS* or *FAIL*) of the operation
code	attribute	string	optional	Error code for failure condition.
Text	element	TextList	optional	Error text for failure condition.
IDs	element	UniqueIDList	optional	Activity ID from requesting system, and if successful, the paired activity ID from the receiving system.

9.2.3 Sample Message

[*Request*]

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#CreateActivity</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:SPASOFT</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:261335f2-bd1d-49e6-96b4-8fcdf2137f8a</wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.micros.com/HTNGActivity/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-d99e21bf-1227-410c-bea3-d8705adcf032">
                <wsu:Created>2006-11-02T20:22:14Z</wsu:Created>
                <wsu:Expires>2006-11-02T20:27:14Z</wsu:Expires>
            </wsu:Timestamp>
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
6f2c0e7f-ac98-4ea3-a726-d67076fc9c72">
                <wsse:Username>HTNG</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">HTNG</wsse:Password>
                <wsse:Nonce>IWGqOAykiUQXnWN5x1wS9Q==</wsse:Nonce>
                <wsu:Created>2006-11-02T20:22:14Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
```

```
    </soap:Header>
    <soap:Body>
        <CreateActivityRequest xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">
            <ResortId>HTNG1</ResortId>
            <ReservationHeader>
                <ReservationID source="OPERA">1620979</ReservationID>
                <TimeSpan>
                    <Start>2006-11-03T09:15:00</Start>
                    <End>2006-11-03T09:45:00</End>
                </TimeSpan>
            </ReservationHeader>
            <Activities>
                <Activity status="BOOK">
                    <ActivityIDs>
                        <UniqueID xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">3633</UniqueID>
                    </ActivityIDs>
                    <ActivityType>Spa</ActivityType>
                    <Location>Tranquility</Location>
                    <Name language="en">Tranquility</Name>
                    <NumberOfPersons>1</NumberOfPersons>
                    <TimeSpan>
                        <Start>2006-11-03T09:15:00</Start>
                        <End>2006-11-03T09:45:00</End>
                    </TimeSpan>
                    <Duration>
                        <Start>2006-11-03T09:15:00</Start>
                        <End>2006-11-03T09:45:00</End>
                    </Duration>
                    <Description language="en">Botanical Bath</Description>
                    <Amount currencyCode="US">        45.00</Amount>
                    <Extensions>
                        <ActivitySpecific>Provider: Cook Evelyn</ActivitySpecific>
                        <ActivitySpecific>Facility: Hydrotherapy Tub 1</ActivitySpecific>
                        <ActivitySpecific>ReservationNo: R1185</ActivitySpecific>
                    </Extensions>
                </Activity>
            </Activities>
        </CreateActivityRequest>
    </soap:Body>
</soap:Envelope>
```


[*Response*]


```
<?xml version="1.0" encoding="utf-8"?>
```

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#CreateActivityResponse</wsa:Action>
        <wsa:MessageID>urn:uuid:114edffa-4634-49fe-93af-a0431ab0f46b</wsa:MessageID>
        <wsa:RelatesTo>urn:uuid:261335f2-bd1d-49e6-96b4-8fcdf2137f8a</wsa:RelatesTo>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-4f4401bc-e095-4db8-8f34-eb53bd8a656f">
                <wsu:Created>2006-11-02T20:20:52Z</wsu:Created>
                <wsu:Expires>2006-11-02T20:25:52Z</wsu:Expires>
            </wsu:Timestamp>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <CreateActivityResponse xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types">
            <Results resultStatusFlag="SUCCESS">
                <IDs xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">
                    <UniqueID source="OPERA">81453</UniqueID>
                    <UniqueID source="SPASOFT">3633</UniqueID>
                </IDs>
            </Results>
        </CreateActivityResponse>
    </soap:Body>
</soap:Envelope>
```


9.3 Update Activity

The update activity message is used to modify an existing activity record in an external system. The requesting system submits an UpdateActivityRequest (ActivityRequest) message, and the responding system returns and UpdateActivityResponse (ActivityResponse). This message is equivalent to the Create Activity message, except that the receiving system may optionally return an error if the activity is not found. Alternatively, the receiving system may simply create the activity as if it was a new request.

Port	ActivityPortType
Binding	ActivityBinding
Operation	UpdateActivity
Soap Action	http://htng.org/PWSWG/2006/08/SingleGuestItinerary#UpdateActivity
Input	UpdateActivityRequest
Output	UpdateActivityResponse
Primary Schema	Activity.xsd
Role(s) Implemented	Itinerary Consolidation System, Activity Reservation System

9.3.1 Sample Message

[*Request*]

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#UpdateActivity</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:SPASOFT</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:3661c78a-2d08-4a83-bc9c-983389fbaca2</wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.micros.com/HTNGActivity/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-1fbf61db-265b-479b-9561-bde2cf5f9fe9">
                <wsu:Created>2006-11-02T20:24:58Z</wsu:Created>
                <wsu:Expires>2006-11-02T20:29:58Z</wsu:Expires>
            </wsu:Timestamp>
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
64afe864-ae86-4380-bf21-ab5b5cb3b81e">
                <wsse:Username>HTNG</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">HTNG</wsse:Password>
                <wsse:Nonce>2ipy/inlnw0dPBS1isvpwA==</wsse:Nonce>
                <wsu:Created>2006-11-02T20:24:58Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <UpdateActivityRequest xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">
            <ResortId>HTNG1</ResortId>
            <ReservationHeader>
                <ReservationID source="OPERA">1621230</ReservationID>
                <TimeSpan>
                    <Start>2006-11-03T18:30:00</Start>
                    <End>2006-11-03T20:00:00</End>
                </TimeSpan>
            </ReservationHeader>
            <Activities>
                <Activity status="BOOK">
```

```
                <ActivityIDs>
                    <UniqueID xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">3638</UniqueID>
                </ActivityIDs>
                <ActivityType>Spa</ActivityType>
                <Location>Tranquility</Location>
                <Name language="en">Tranquility</Name>
                <NumberOfPersons>1</NumberOfPersons>
                <TimeSpan>
                    <Start>2006-11-03T18:30:00</Start>
                    <End>2006-11-03T20:00:00</End>
                </TimeSpan>
                <Duration>
                    <Start>2006-11-03T18:30:00</Start>
                    <End>2006-11-03T20:00:00</End>
                </Duration>
                <Description language="en">Thai Massage 80 Minutes</Description>
                <Amount currencyCode="US">        150.00</Amount>
                <Extensions>
                    <ActivitySpecific>Provider: Farrows Pierre</ActivitySpecific>
                    <ActivitySpecific>Facility: Massage 1</ActivitySpecific>
                    <ActivitySpecific>ReservationNo: R1190</ActivitySpecific>
                </Extensions>
            </Activity>
        </Activities>
    </UpdateActivityRequest>
  </soap:Body>
</soap:Envelope>
```


[*Response*]


```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
  <soap:Header>
    <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#UpdateActivityResponse</wsa:Action>
    <wsa:MessageID>urn:uuid:9f534e5b-ef7e-4468-9599-07fee7dd823f</wsa:MessageID>
    <wsa:RelatesTo>urn:uuid:3661c78a-2d08-4a83-bc9c-983389fbaca2</wsa:RelatesTo>
    <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
    <wsse:Security>
      <wsu:Timestamp wsu:Id="Timestamp-bca32201-b0a0-4d36-b51d-a98639eae83d">
        <wsu:Created>2006-11-02T20:23:31Z</wsu:Created>
        <wsu:Expires>2006-11-02T20:28:31Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
```

```
    </soap:Header>
    <soap:Body>
        <UpdateActivityResponse xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types">
            <Results resultStatusFlag="SUCCESS"/>
        </UpdateActivityResponse>
    </soap:Body>
</soap:Envelope>
```

9.4 Cancel Activity

The cancel activity message is used to remove an existing activity record from an external system. The requesting system submits a CancelActivityRequest (ActivityRequest) message, and the responding system returns and CancelActivityResponse (ActivityResponse). This message is also equivalent to the Create Activity message, except that the Activity records defined in the ActivityList need only include minimal information to uniquely identify the item(s) to be cancelled.

Port	ActivityPortType
Binding	ActivityBinding
Operation	CancelActivity
Soap Action	http://htng.org/PWSWG/2006/08/SingleGuestItinerary#CancelActivity
Input	CancelActivityRequest
Output	CancelActivityResponse
Primary Schema	Activity.xsd
Role(s) Implemented	Itinerary Consolidation System, Activity Reservation System

9.4.1 Sample Message

[*Request*]

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#CancelActivity</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:SPASOFT</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:55222e43-01ea-4a15-ac6e-49bbb868eeaf</wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.micros.com/HTNGActivity/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-8098078c-028f-4b13-ace6-7ad95a75900a">
                <wsu:Created>2006-11-02T20:30:57Z</wsu:Created>
                <wsu:Expires>2006-11-02T20:35:57Z</wsu:Expires>
```

```
            </wsu:Timestamp>
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
14690222-cceb-4993-984c-a6b3e8c49def">
                <wsse:Username>HTNG</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">HTNG</wsse:Password>
                <wsse:Nonce>cZjtNnAd24gHWfZEhuEDHQ==</wsse:Nonce>
                <wsu:Created>2006-11-02T20:30:57Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <CancelActivityRequest xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">
            <ResortId>HTNG1</ResortId>
            <ReservationHeader>
                <ReservationID source="OPERA">1621230</ReservationID>
                <TimeSpan>
                    <Start>2006-11-03T18:30:00</Start>
                    <End>2006-11-03T20:00:00</End>
                </TimeSpan>
            </ReservationHeader>
            <Activities>
                <Activity status="CANC">
                    <ActivityIDs>
                        <UniqueID xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">3638</UniqueID>
                    </ActivityIDs>
                    <ActivityType>Spa</ActivityType>
                    <Location>Tranquility</Location>
                    <Name language="en">Tranquility</Name>
                    <NumberOfPersons>1</NumberOfPersons>
                    <TimeSpan>
                        <Start>2006-11-03T18:30:00</Start>
                        <End>2006-11-03T20:00:00</End>
                    </TimeSpan>
                    <Duration>
                        <Start>2006-11-03T18:30:00</Start>
                        <End>2006-11-03T20:00:00</End>
                    </Duration>
                    <Description language="en">Thai Massage 80 Minutes</Description>
                    <Amount currencyCode="US">        150.00</Amount>
                    <Extensions>
                        <ActivitySpecific>Provider: Farrows Pierre</ActivitySpecific>
                        <ActivitySpecific>Facility: Massage 1</ActivitySpecific>
                        <ActivitySpecific>ReservationNo: R1190</ActivitySpecific>
```

```
                </Extensions>
            </Activity>
        </Activities>
    </CancelActivityRequest>
  </soap:Body>
</soap:Envelope>
```

[*Response*]

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
  <soap:Header>
    <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#CancelActivityResponse</wsa:Action>
    <wsa:MessageID>urn:uuid:6a8d39cc-293a-43c8-bcb1-0c7ab7306bfc</wsa:MessageID>
    <wsa:RelatesTo>urn:uuid:55222e43-01ea-4a15-ac6e-49bbb868eeaf</wsa:RelatesTo>
    <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
    <wsse:Security>
      <wsu:Timestamp wsu:Id="Timestamp-74887dea-6abb-46bf-9191-b05595d882e9">
        <wsu:Created>2006-11-02T20:29:30Z</wsu:Created>
        <wsu:Expires>2006-11-02T20:34:30Z</wsu:Expires>
      </wsu:Timestamp>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <CancelActivityResponse xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types">
      <Results resultStatusFlag="SUCCESS"/>
    </CancelActivityResponse>
  </soap:Body>
</soap:Envelope>
```

Chapter 10 Activity Provider Port Type

The Activity Provider Port defines two functions which are implemented only by the Activity Reservation System. These are

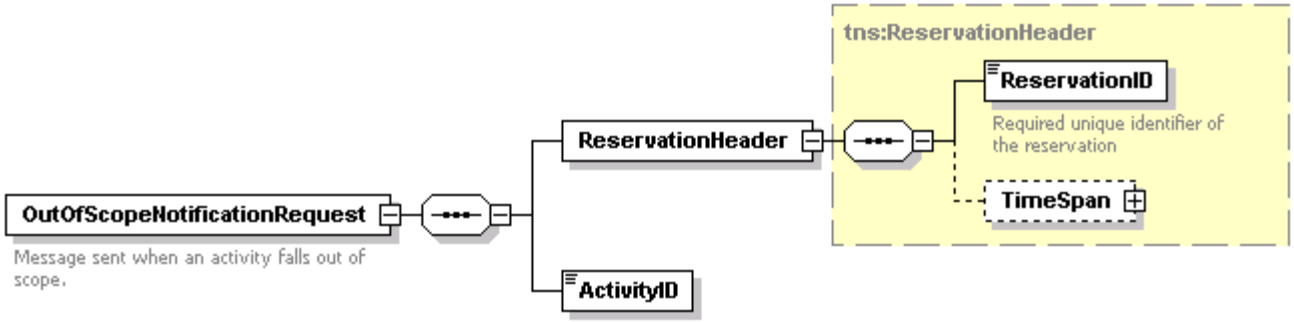
- OutOfScopeNotification
  - Function to notify the Activity Reservation System of a change in scope for a scheduled activity.
- FetchActivities
  - Function to fetch a set of activities for a specific guest profile or reservation.

10.1 Out of Scope Notification

The Out of Scope Notification is a message designed primarily to inform the Activity Reservation System of a change in scope for a guest reservation which an activity is linked to. For instance, if an activity is linked to a guest reservation in the Accommodation Reservation System, and the reservation is cancelled, the Activity Reservation System should be notified as such. Furthermore, if a reservation date span is changed that formerly included an activity; the Activity Reservation System would likely need to be notified.

Notification is not required by the Accommodation Reservation System for a change in reservation scope. There are typically three possible actions: Request cancellation in the Activity Reservation System, notify the Activity Reservation System, or take no action. The choice is typically a business operational policy, and is therefore often provided as a configuration option within the Accommodation Reservation System.

10.1.1 OutOfScopeNotificationRequest



OutOfScopeNotificationRequest
<xs:element name="OutOfScopeNotificationRequest">
<xs:annotation>
<xs:documentation>Message sent when an activity falls out of scope.</xs:documentation>
</xs:annotation>
<xs:complexType>

```
        <xs:sequence>
            <xs:element name="ReservationHeader">
                <xs:complexType>
                    <xs:complexContent mixed="false">
                        <xs:extension xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types" base="q1:ReservationHeader">
                            <xs:attribute name="reservationStatus" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
                                type="q2:ReservationStatusType" use="optional" />
                            <xs:attribute name="otherReservationStatus" type="xs:string" use="optional" />
                        </xs:extension>
                    </xs:complexContent>
                </xs:complexType>
            </xs:element>
            <xs:element name="ActivityID" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q3:UniqueID" />
        </xs:sequence>
        <xs:attribute name="outOfScopeAction" xmlns:q4="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
            type="q4:OutOfScopeActionType" use="required" />
        <xs:attribute name="otherOutOfScopeAction" type="xs:string" use="optional" />
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
outOfScopeAction	attribute	OutOfScopeActionType	required	One of *RELEASE*, *BREAK*, *NOTIFICATION*, or *OTHER*. *RELEASE* is used to request that the activity be deleted. *NOTIFICATION* is sent to indicate a notification only. The remaining two values are not currently defined.
otherOutOfScopeAction	attribute	string	optional	Not used.
ReservationHeader	element		required	Identifies the guest's reservation applicable to the notification.
ActivityID	element	UniqueID	required	Identifies the scheduled activity related to the notification.

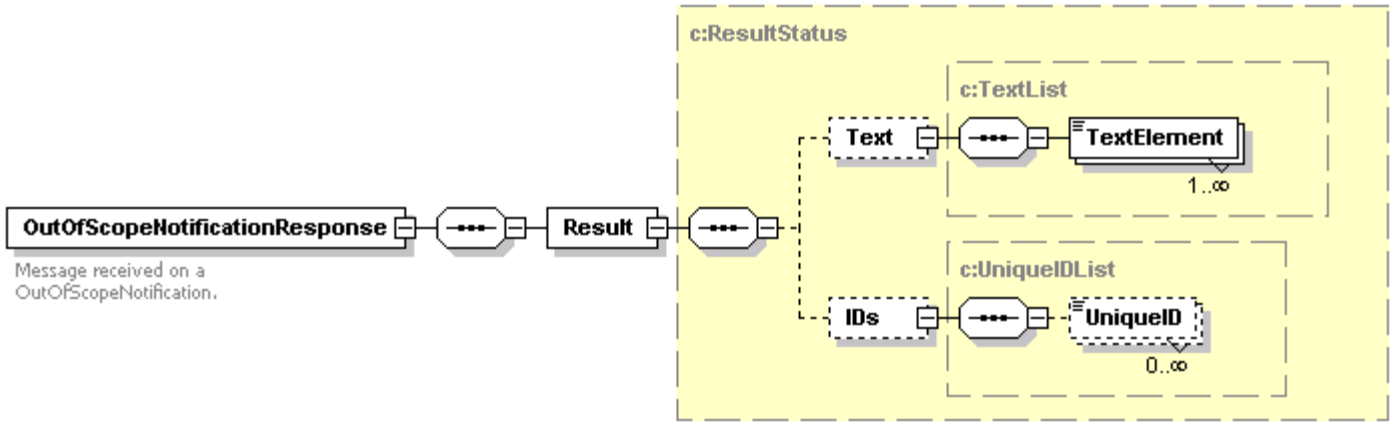
ReservationHeader

ReservationHeader

```
<xs:complexType name="ReservationHeader">
    <xs:annotation>
    <xs:documentation>Used to uniquely identify a reservation using it's internal ID, it's begin date and duration or end date.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="ReservationID" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueID">
            <xs:annotation>
            <xs:documentation>Required unique identifier of the reservation</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="TimeSpan" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
            type="q2:OGTimeSpan" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
ReservationID	element	UniqueID	required	The reservation ID as defined by the notifying system.
TimeSpan	element	OGTimeSpan	optional	The time span of the reservation.

10.1.2 OutOfScopeNotificationResponse



OutOfScopeNotificationResponse

```
<xs:element name="OutOfScopeNotificationResponse">
    <xs:annotation>
    <xs:documentation>Message received on a OutOfScopeNotification.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Result" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:ResultStatus" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
Result	element	ResultStatus	required	The result from the notification. No action is typically required on the result object by the initiating system.

10.1.3 Sample Message

[*Request*]

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
```

```
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#OutOfScopeNotification</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:OPERA</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:33be344a-b857-4601-a64b-6f80acbac94b</wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.springermiller.com/HTNG_2_0/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-abb75372-7ae1-4810-a0cd-f311c5f5e79e">
                <wsu:Created>2007-02-05T18:06:40Z</wsu:Created>
                <wsu:Expires>2007-02-05T18:11:40Z</wsu:Expires>
            </wsu:Timestamp>
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
166f2cf0-82ea-4aed-b2e4-c716c22ff737">
                <wsse:Username>OPERA</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">OPERA</wsse:Password>
                <wsse:Nonce>KnHeQA1M7kYQRVluo6Gjrw==</wsse:Nonce>
                <wsu:Created>2007-02-05T18:06:40Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <OutOfScopeNotificationRequest xmlns:r="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
xmlns:n="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
xmlns:a="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
xmlns:p="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types" outOfScopeAction="RELEASE"
xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types">
            <a:ReservationHeader reservationStatus="CANCELLED">
                <a:ReservationID source="OPERA">988386</a:ReservationID>
            </a:ReservationHeader>
            <a:ActivityID source="SPASOFT">3741</a:ActivityID>
        </OutOfScopeNotificationRequest>
    </soap:Body>
</soap:Envelope>
```


[*Response*]


```
<?xml version="1.0" encoding="utf-8"?>
```

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#OutOfScopeNotificationResponse</wsa:Action>
        <wsa:MessageID>urn:uuid:bf7d32d7-e340-4d88-86dc-fa0a9fb5d558</wsa:MessageID>
        <wsa:RelatesTo>urn:uuid:33be344a-b857-4601-a64b-6f80acbac94b</wsa:RelatesTo>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-04f51b7d-ac10-4a52-9a12-d660353755ff">
                <wsu:Created>2007-02-05T18:08:10Z</wsu:Created>
                <wsu:Expires>2007-02-05T18:13:10Z</wsu:Expires>
            </wsu:Timestamp>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <OutOfScopeNotificationResponse xmlns:r="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
xmlns:a="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types">
            <a:Result code="" resultStatusFlag="SUCCESS">
                <c:Text>
                    <c:TextElement language="en"/>
                </c:Text>
                <c:IDs>
                    <c:UniqueID source="SPASOFT">3741</c:UniqueID>
                </c:IDs>
            </a:Result>
        </OutOfScopeNotificationResponse>
    </soap:Body>
</soap:Envelope>
```

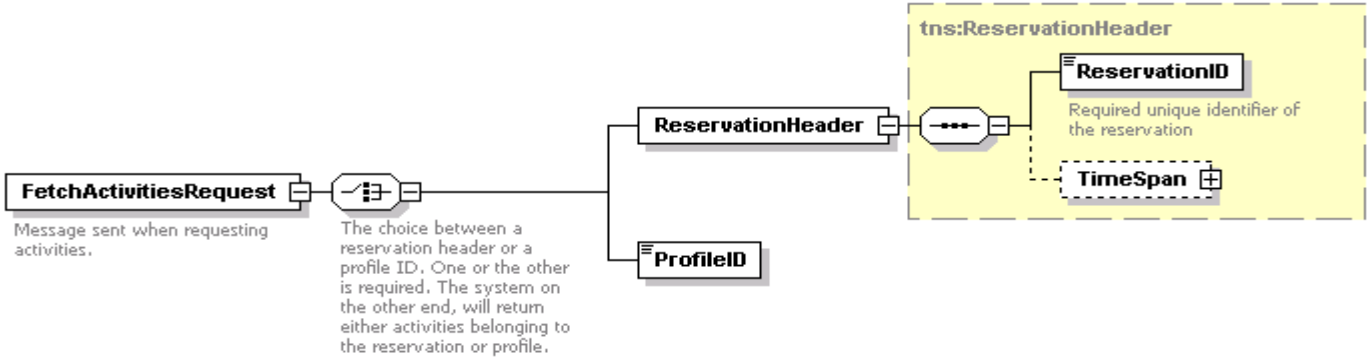

10.2 Fetch Activities

The Fetch Activities message is provided by the Activity Reservation System in order for an external system (typically the Activity Consolidation System) to query all scheduled activities for a particular guest. The request includes a choice element either indicating a guest profile ID or reservation ID, as defined in the receiving system.

Port	ActivityProviderPortType
Binding	ActivityProviderBinding
Operation	FetchActivities
Soap Action	http://htng.org/PWSWG/2006/08/SingleGuestItinerary#FetchActivities
Input	FetchActivitiesRequest
Output	FetchActivitiesResponse

Primary Schema	Activity.xsd
Role(s) Implemented	Activity Reservation System

10.2.1 FetchActivitiesRequest



FetchActivitiesRequest

```
<xs:element name="FetchActivitiesRequest">
    <xs:annotation>
    <xs:documentation>Message sent when requesting activities.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:choice>
            <xs:annotation>
            <xs:documentation xml:lang="en">The choice between a reservation header or a profile ID. One or the other is required. The system on the other
                end, will return either activities belonging to the reservation or profile.</xs:documentation>
            </xs:annotation>
            <xs:element name="ReservationHeader" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
                type="q1:ReservationHeader" />
            <xs:element name="ProfileID" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q2:UniqueID" />
        </xs:choice>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
ReservationHeader	element	ReservationHeader	required	Identifies the reservation to fetch activities for. Either this element or the ProfileID must be sent.
ProfileID	element	UniqueID	required	Identifies the guest profile ID to fetch activities for. Either this element

				of the ReservationHeader must be sent.

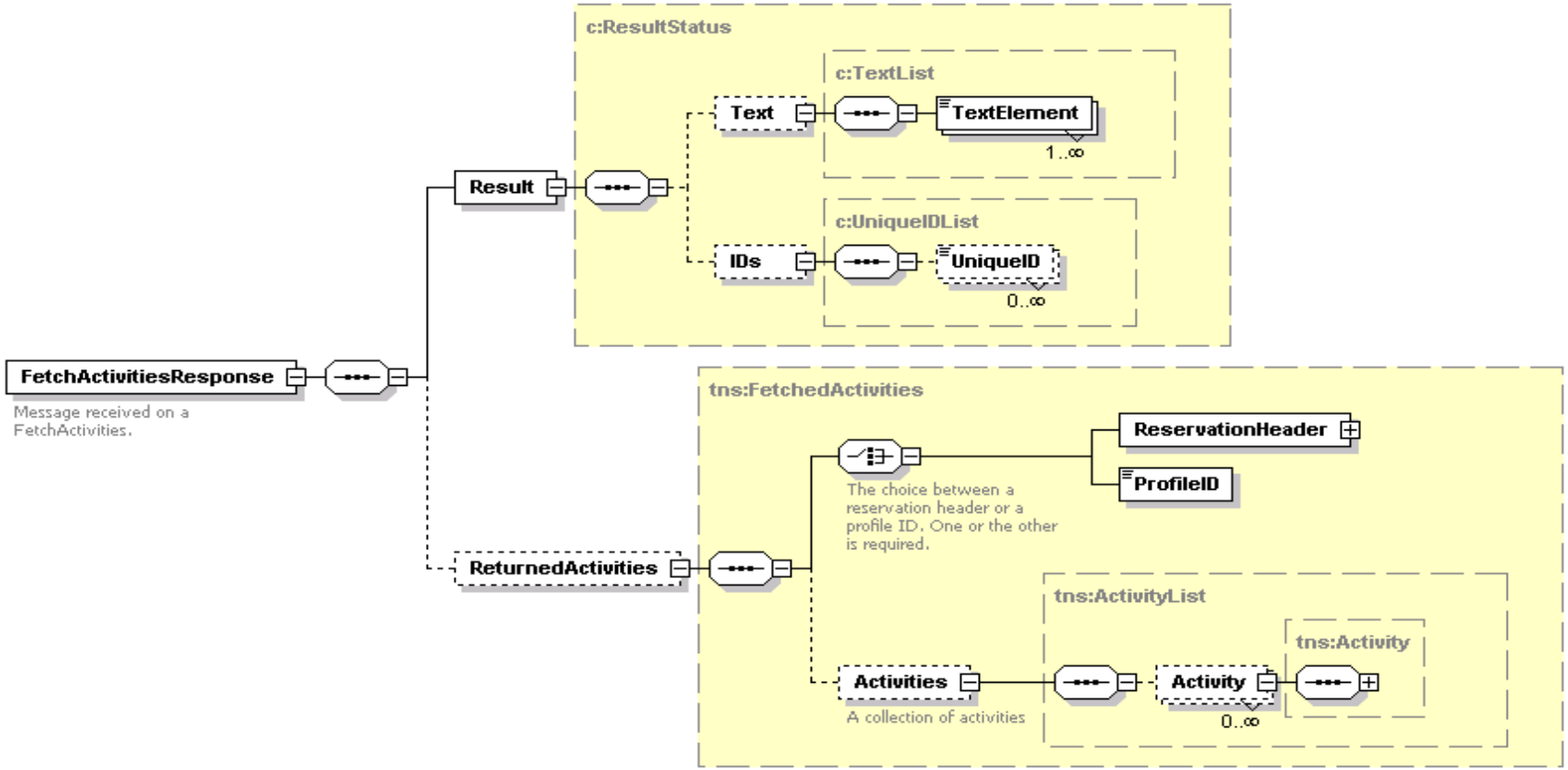
ReservationHeader

ReservationHeader

```
<xs:complexType name="ReservationHeader">
    <xs:annotation>
    <xs:documentation>Used to uniquely identify a reservation using it's internal ID, it's begin date and duration or end date.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="ReservationID" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueID">
            <xs:annotation>
            <xs:documentation>Required unique identifier of the reservation</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="TimeSpan" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
            type="q2:OGTimeSpan" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
ReservationID	element	UniqueID	required	The ID as defined in the receiving system.
TimeSpan	element	OGTimeSpan	optional	The date and duration of the reservation.

10.2.2 FetchActivitiesResponse



FetchActivitiesResponse

```
<xs:element name="FetchActivitiesResponse">
    <xs:annotation>
    <xs:documentation>Message received on a FetchActivities.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Result" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:ResultStatus" />
            <xs:element minOccurs="0" name="ReturnedActivities" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
                type="q2:FetchedActivities" />
        </xs:sequence>
```

```
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
Result	element	ResultStatus	required	The result of the operation.
ReturnedActivities	element	FetchedActivities	optional	A list of activities for the requested guest or reservation.

FetchedActivities

FetchedActivities

```
<xs:complexType name="FetchedActivities">
    <xs:annotation>
    <xs:documentation>Contains either a reservation header or ProfileID, indicating the owner, plus a collection of activities.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:choice>
            <xs:annotation>
            <xs:documentation xml:lang="en">The choice between a reservation header or a profile ID. One or the other is required.</xs:documentation>
            </xs:annotation>
            <xs:element name="ReservationHeader" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
                type="q1:ReservationHeader" />
            <xs:element name="ProfileID" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q2:UniqueID" />
        </xs:choice>
        <xs:element minOccurs="0" name="Activities" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
            type="q3:ActivityList">
            <xs:annotation>
            <xs:documentation>A collection of activities</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
ReservationHeader	element	ReservationHeade r	required	The reservation header (if sent in the request).
ProfileID	element	UniqueID	required	The profile ID (if send in the request).
Activities	element	ActivityList	optional	A collection of Activity records.

10.2.3 Sample Message

[*Request*]

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
```

```
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#FetchActivities</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:OPERA</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:46459d60-97da-42cf-b7de-529251c84421</wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.springermiller.com/HTNG_2_0/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-5fabacfd-fec2-493e-b21c-611a6f035ef4">
                <wsu:Created>2007-01-25T18:57:33Z</wsu:Created>
                <wsu:Expires>2008-01-25T19:02:33Z</wsu:Expires>
            </wsu:Timestamp>
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
998fd48c-d649-4837-abb6-3109483a976e">
                <wsse:Username>OPERA</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">OPERA</wsse:Password>
                <wsse:Nonce>GNXdMfR1op/QbjurLlO7SQ==</wsse:Nonce>
                <wsu:Created>2007-01-25T18:57:33Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <FetchActivitiesRequest xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <ReservationHeader>
                <ReservationID>988386</ReservationID>
            </ReservationHeader>
        </FetchActivitiesRequest>
    </soap:Body>
</soap:Envelope>
```


[*Response*]


```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#FetchActivitiesResponse</wsa:Action>
        <wsa:MessageID>urn:uuid:08f381d8-584f-4ef5-aedf-32c171a62f8f</wsa:MessageID>
        <wsa:RelatesTo>urn:uuid:46459d60-97da-42cf-b7de-529251c84421</wsa:RelatesTo>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
```

```
        <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-36d3acd5-d1e4-4e60-b3e0-8da4ed2c4dfc">
                <wsu:Created>2007-02-05T19:19:31Z</wsu:Created>
                <wsu:Expires>2007-02-05T19:24:31Z</wsu:Expires>
            </wsu:Timestamp>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <FetchActivitiesResponse xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">
            <Result code="" resultStatusFlag="SUCCESS">
                <c:Text>
                    <c:TextElement language="en"/>
                </c:Text>
                <c:IDs>
                    <c:UniqueID source="SPASOFT">3741</c:UniqueID>
                    <c:UniqueID source="SPASOFT">4510</c:UniqueID>
                </c:IDs>
            </Result>
            <ReturnedActivities>
                <ReservationHeader>
                    <ReservationID source="OPERA">988386</ReservationID>
                    <TimeSpan>
                        <Start>2007-01-25T09:30:00</Start>
                        <End>2007-01-29T13:00:00</End>
                    </TimeSpan>
                </ReservationHeader>
                <Activities>
                    <Activity status="HOLD">
                        <ActivityIDs>
                            <c:UniqueID source="SPASOFT">3741</c:UniqueID>
                        </ActivityIDs>
                        <ActivityType>Spa</ActivityType>
                        <Location>Tranquility Spa, Palm Springs</Location>
                        <Name language="en">Tranquility Spa, Palm Springs</Name>
                        <NumberOfPersons>1</NumberOfPersons>
                        <TimeSpan>
                            <Start>2007-01-25T09:30:00</Start>
                            <End>2007-01-25T10:00:00</End>
                        </TimeSpan>
                        <Duration>
                            <Start>2007-01-25T09:30:00</Start>
                            <End>2007-01-25T10:00:00</End>
                        </Duration>
                        <Description language="en">Champagne Bath</Description>
```

```
                <Amount currencyCode="US"/>
                <Extensions>
                    <ActivitySpecific>Provider: Edwards Serena</ActivitySpecific>
                    <ActivitySpecific>Facility: Hydrotherapy Tub 2</ActivitySpecific>
                    <ActivitySpecific>ReservationNo: R1275</ActivitySpecific>
                </Extensions>
            </Activity>
            <Activity status="BOOK">
                <ActivityIDs>
                    <c:UniqueID source="SPASOFT">4510</c:UniqueID>
                </ActivityIDs>
                <ActivityType>Spa</ActivityType>
                <Location>Tranquility Spa, Palm Springs</Location>
                <Name language="en">Tranquility Spa, Palm Springs</Name>
                <NumberOfPersons>1</NumberOfPersons>
                <TimeSpan>
                    <Start>2007-01-29T12:00:00</Start>
                    <End>2007-01-29T13:00:00</End>
                </TimeSpan>
                <Duration>
                    <Start>2007-01-29T12:00:00</Start>
                    <End>2007-01-29T13:00:00</End>
                </Duration>
                <Description language="en">European Facial</Description>
                <Amount currencyCode="US"/>
                <Extensions>
                    <ActivitySpecific>Provider: Evans Andrea</ActivitySpecific>
                    <ActivitySpecific>Facility: Facial Room 5</ActivitySpecific>
                    <ActivitySpecific>ReservationNo: R1464</ActivitySpecific>
                </Extensions>
            </Activity>
        </Activities>
    </ReturnedActivities>
  </FetchActivitiesResponse>
 </soap:Body>
</soap:Envelope>
```

Chapter 11 Activity Sync Port Type

The Activity Sync port defines a single function which may be called from the Activity Reservation System in order to notify the Itinerary Consolidation System of a change in the location of an activity.

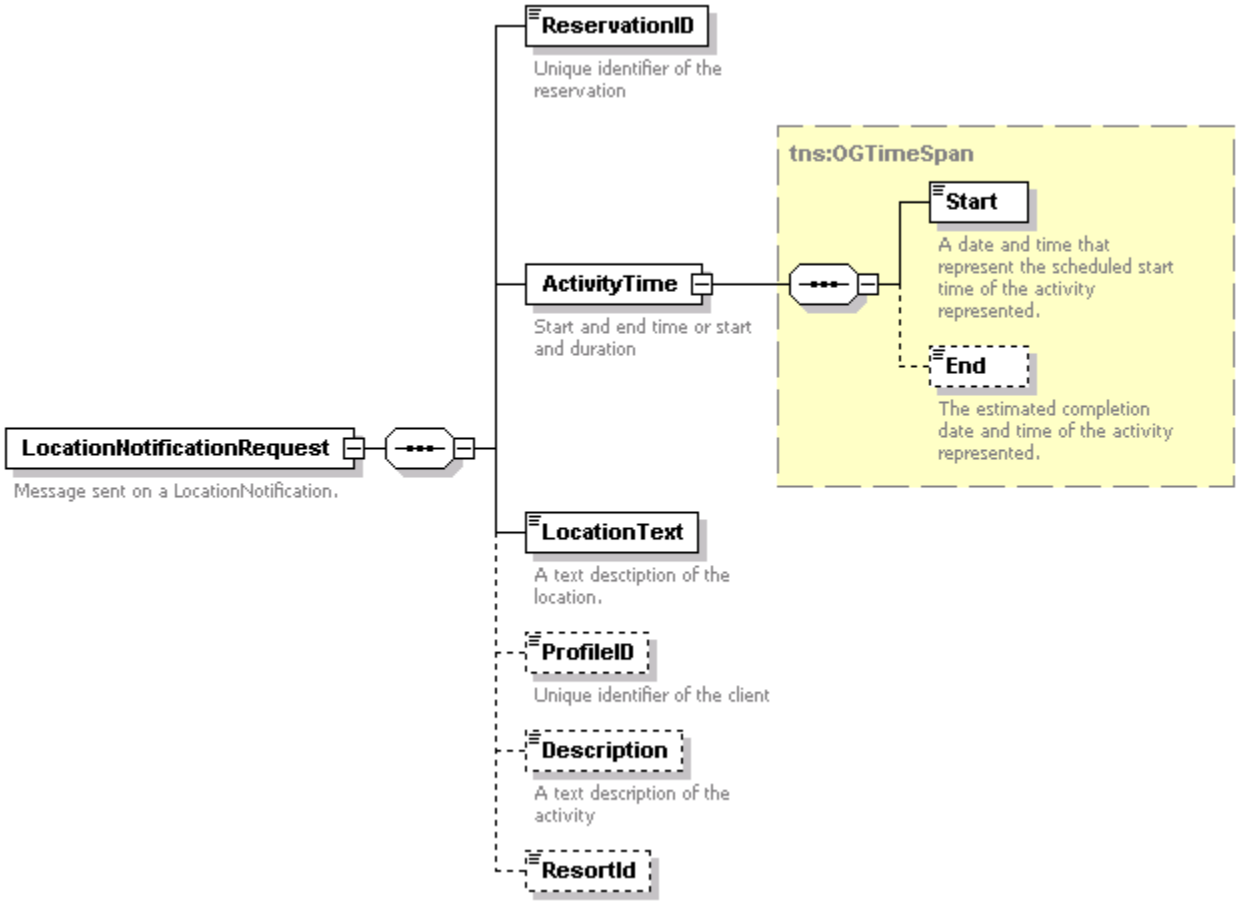
- LocationNotification
  - Function called from the Activity Reservation System in order to notify a change in the location for a scheduled activity.

11.1 Location Notification

A Location Notification message is submitted by the Activity Reservation System in order to notifity the Accommodation Reservation System of a guest's current physical location. Some reservation systems implement this feature in order to locate a guest in order to forward phone calls or deliver messages.

Port	ActivitySyncPortType
Binding	ActivitySyncBinding
Operation	ActivityLookup
Soap Action	http://htng.org/PWSWG/2006/08/SingleGuestItinerary#LocationNotification
Input	LocationNotificationRequest
Output	LocationNotificationResponse
Primary Schema	Activity.xsd
Role(s) Implemented	Itinerary Consolidation System

11.1.1 LocationNotificationRequest



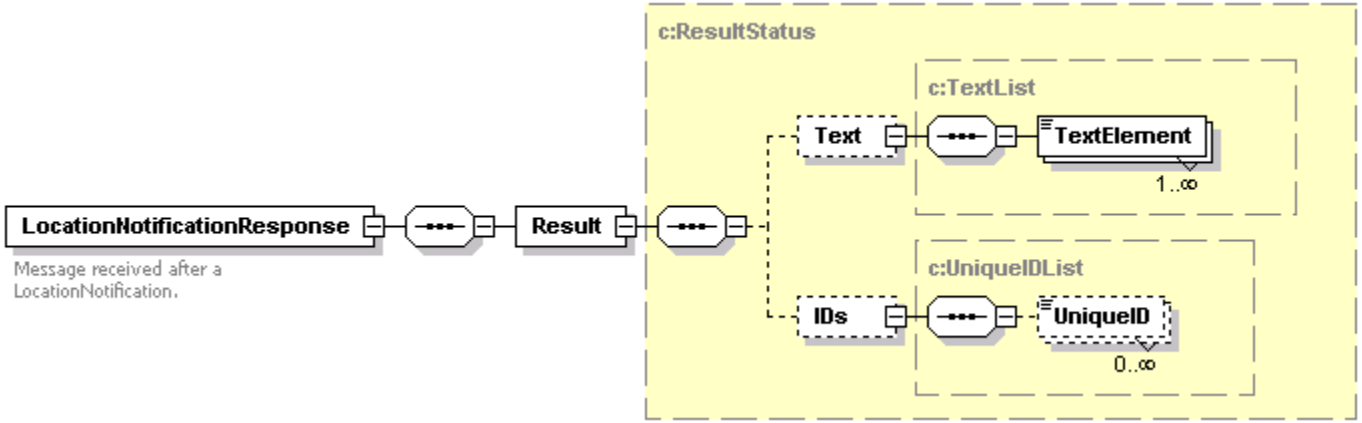
LocationNotificationRequest

```
<xs:element name="LocationNotificationRequest">
    <xs:annotation>
    <xs:documentation>Message sent on a LocationNotification.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ReservationID" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueID">
```

```
            <xs:annotation>
            <xs:documentation>Unique identifier of the reservation</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="ActivityTime" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types" type="q2:OGTimeSpan">
            <xs:annotation>
            <xs:documentation>Start and end time or start and duration</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="LocationText" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q3:Text">
            <xs:annotation>
            <xs:documentation>A text desctiption of the location.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="ProfileID" xmlns:q4="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q4:UniqueID">
            <xs:annotation>
            <xs:documentation>Unique identifier of the client</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="Description" xmlns:q5="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q5:Text">
            <xs:annotation>
            <xs:documentation>A text description of the activity</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="ResortId" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="locationNotificationStatus" xmlns:q6="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
        type="q6:LocationNotificationStatus" use="required" />
    <xs:attribute name="otherLocationNotificationStatus" type="xs:string" use="optional" />
  </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
locationNotificationStatus	attribute	LocationNotificationStatus	required	One of *SET*, *CLEAR*, or *OTHER*.
otherLocationNotificationStatus	attribute	string	optional	Not used.
ReservationID	element	UniqueID	required	The reservation ID as defined in the receiving system.
ActivityTime	element	OGTimeSpan	required	The activity time.
LocationText	element	Text	required	A description of the guest's location.
ProfileID	element	UniqueID	optional	The ID of the guest record as defined in the receiving system.
Description	element	Text	optional	The description of the activity.
ResortId	element	string	optional	The property ID.

11.1.2 LocationNotificationResponse



LocationNotificationResponse				
<xs:element name="LocationNotificationResponse">
<xs:annotation>
<xs:documentation>Message received after a LocationNotification.</xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:sequence>
<xs:element name="Result" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:ResultStatus" />
</xs:sequence>
</xs:complexType>
</xs:element>				
Name	**Type**	**Data Type**	**Use**	**Comments**
Result	element	ResultStatus	required	The result of the notification.

11.1.3 Sample Message

[*Request*]

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#LocationNotification</wsa:Action>
        <wsa:From>
```

```
            <wsa:Address>urn:SPASOFT</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:5518896e-eed9-4a8b-89b6-4c4a5a3f756f</wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.micros.com/HTNGActivity/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-aeadcfcd-62bb-4569-9adf-2058cfd1f130">
                <wsu:Created>2006-11-08T17:17:45Z</wsu:Created>
                <wsu:Expires>2008-11-08T17:22:45Z</wsu:Expires>
            </wsu:Timestamp>
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
3bdd620a-3146-4012-8da3-a202e5b1f896">
                <wsse:Username>HTNG</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">HTNG</wsse:Password>
                <wsse:Nonce>rtQ+bYWTWfGrRipi6fhngg==</wsse:Nonce>
                <wsu:Created>2006-11-08T17:17:45Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <LocationNotificationRequest xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types D:\htng\Activity.xsd" locationNotificationStatus="SET"
otherLocationNotificationStatus="OCEAN">
            <ReservationID source="OPERA">1621229</ReservationID>
            <ActivityTime>
                <Start>2006-11-07T09:30:47.0Z</Start>
                <End>2006-11-07T09:30:47.0Z</End>
            </ActivityTime>
            <LocationText language="en-us">OCEAN</LocationText>
            <ProfileID source="OPERA">450018</ProfileID>
            <Description language="en-us">OCEAN</Description>
            <ResortId>HTNG1</ResortId>
        </LocationNotificationRequest>
    </soap:Body>
</soap:Envelope>
```


[*Response*]


```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
```

```
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#LocationNotificationResponse</wsa:Action>
        <wsa:MessageID>urn:uuid:e189f893-f319-444c-9786-51d26e6a1861</wsa:MessageID>
        <wsa:RelatesTo>urn:uuid:5518896e-eed9-4a8b-89b6-4c4a5a3f756f</wsa:RelatesTo>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-e1fb8d71-564d-42d1-8a76-d4b305b99aab">
                <wsu:Created>2006-11-13T15:31:43Z</wsu:Created>
                <wsu:Expires>2006-11-13T15:36:43Z</wsu:Expires>
            </wsu:Timestamp>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <LocationNotificationResponse xmlns:r="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
xmlns:n="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
xmlns:a="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
xmlns:p="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types">
            <a:Result resultStatusFlag="SUCCESS"/>
        </LocationNotificationResponse>
    </soap:Body>
</soap:Envelope>
```

Chapter 12 Name Port Type

The Name Port Type defines five functions which provide support for guest profile related operations. In order for notification messages to occur within the interface, a link is typically established for specific guest profile records in both systems. The Name Port includes functions to provide this link. In addition, it is often desired that changes to linked profiles be notified from one (or multiple) systems to the others. The Name Port also provides the means to create a new profile in an external system.

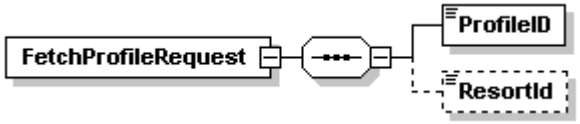
- FetchProfile
  - Request to fetch information for a guest profile based on an external ID
- NewProfile
  - This function may be used to create a new profile in an external system.
- UpdateProfile
  - This function is used to notify an external system when a linked profile is changed.
- ProfileLookup
  - The profile lookup function provides for a generic lookup against several filters.
- Subscription
  - This function is provided for notification back to an external system of the internal ID for a newly created guest profile record.

12.1 Fetch Profile

The fetch profile message is sent in order to request guest details for a specific ID as defined in the external (receiving) system.

Port	NamePortType
Binding	NameBinding
Operation	FetchProfile
Soap Action	http://htng.org/PWSWG/2006/08/SingleGuestItinerary#FetchProfile
Input	FetchProfileRequest
Output	FetchProfileResponse
Primary Schema	Name.xsd
Role(s) Implemented	Customer Profile System, Activity Reservation System

12.1.1 FetchProfileRequest

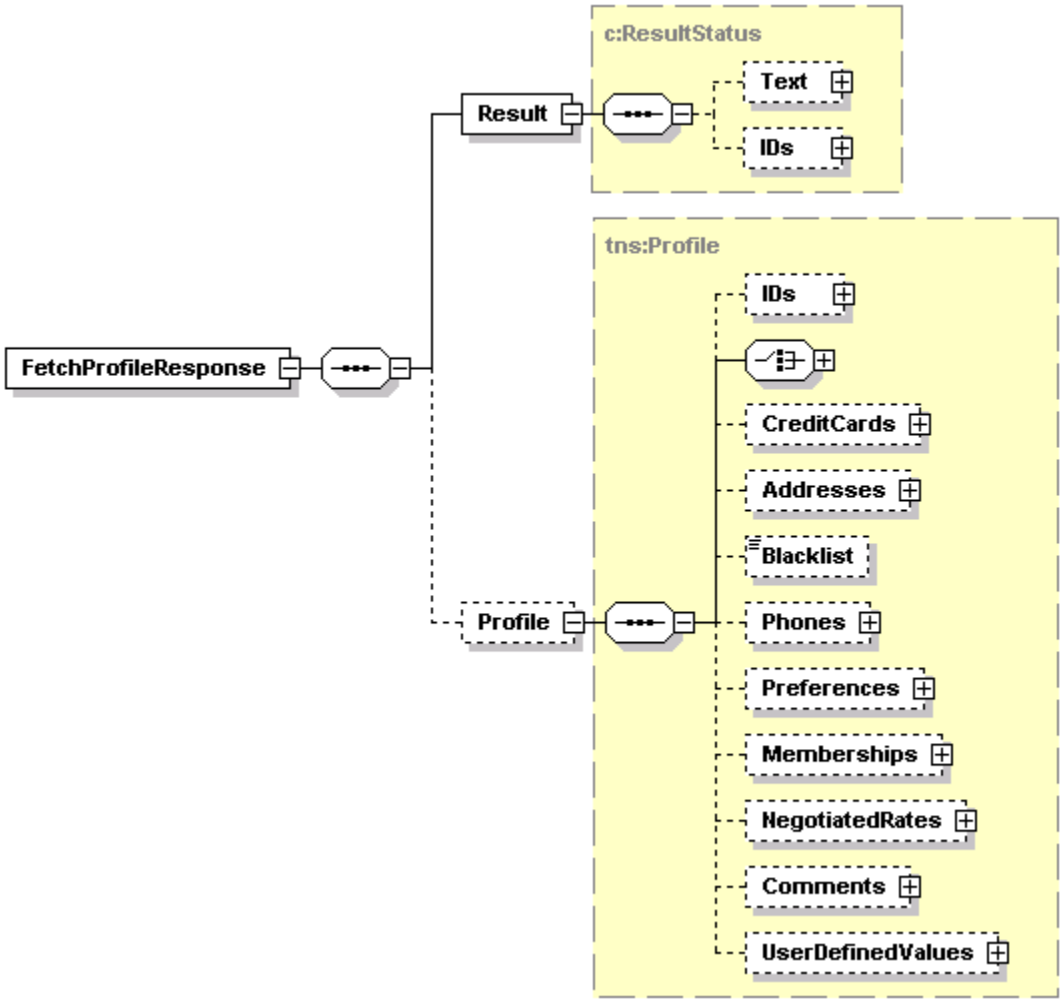


FetchProfileRequest
<xs:element name="FetchProfileRequest">   <xs:complexType>

```
        <xs:sequence>
            <xs:element name="ProfileID" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueID" />
            <xs:element minOccurs="0" name="ResortId" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
ProfileID	element	UniqueID	required	The guest profile ID as defined in the receiving system.
ResortId	element	string	optional	The property the guest is associated with. This field is not required.

12.1.2 FetchProfileResponse



FetchProfileResponse
<xs:element name="FetchProfileResponse"> <xs:complexType>

```
        <xs:sequence>
            <xs:element name="Result" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:ResultStatus" />
            <xs:element minOccurs="0" name="Profile" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q2:Profile" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
Result	element	ResultStatus	required	The result of the operation.
Profile	element	Profile	optional	Guest profile details.

12.1.3 Sample Message

[*Request*]

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#FetchProfile</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:SPASOFT</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:3661c78a-2d08-4a83-bc9c-983389fbaca2</wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.micros.com/HTNGActivity/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-1fbf61db-265b-479b-9561-bde2cf5f9fe9">
                <wsu:Created>2006-11-02T20:24:58Z</wsu:Created>
                <wsu:Expires>2008-11-02T20:29:58Z</wsu:Expires>
            </wsu:Timestamp>
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
64afe864-ae86-4380-bf21-ab5b5cb3b81e">
                <wsse:Username>HTNG</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">HTNG</wsse:Password>
                <wsse:Nonce>2ipy/inlnw0dPBS1isvpwA==</wsse:Nonce>
                <wsu:Created>2006-11-02T20:24:58Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
```

```
        <FetchProfileRequest xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types D:\htng\Name.xsd">
            <ProfileID source="OPERA">450018</ProfileID>
            <ResortId>HTNG1</ResortId>
        </FetchProfileRequest>
    </soap:Body>
</soap:Envelope>
```


[*Response*]


```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#FetchProfileResponse</wsa:Action>
        <wsa:MessageID>urn:uuid:e1f65b7f-5bdc-4566-9c18-9883bdd40a56</wsa:MessageID>
        <wsa:RelatesTo>urn:uuid:3661c78a-2d08-4a83-bc9c-983389fbaca2</wsa:RelatesTo>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-a1c29de1-b460-4e67-9c7d-649c589cd564">
                <wsu:Created>2006-11-13T20:36:03Z</wsu:Created>
                <wsu:Expires>2006-11-13T20:41:03Z</wsu:Expires>
            </wsu:Timestamp>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <FetchProfileResponse xmlns:r="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
xmlns:n="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
xmlns:a="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
xmlns:p="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types">
            <n:Result resultStatusFlag="SUCCESS">
                <c:IDs>
                    <c:UniqueID source="OPERA">450018</c:UniqueID>
                </c:IDs>
            </n:Result>
            <n:Profile nameType="GUEST">
                <n:IDs>
                    <c:UniqueID source="OPERA">450018</c:UniqueID>
                </n:IDs>
                <n:Customer>
                    <n:PersonName familiarName="VANCE">
```

```
                <c:FirstName>TIM</c:FirstName>
                <c:LastName>VANCE</c:LastName>
            </n:PersonName>
        </n:Customer>
        <n:CreditCards>
            <n:NameCreditCard primary="true">
                <c:CardCode>AX</c:CardCode>
                <c:CardHolderName>VANCE/TIM</c:CardHolderName>
                <c:CardNumber>373344556677889</c:CardNumber>
                <c:ExpirationDate>1999-12-31</c:ExpirationDate>
            </n:NameCreditCard>
            <n:NameCreditCard primary="true">
                <c:CardCode>MC</c:CardCode>
                <c:CardHolderName>TIM VANCE</c:CardHolderName>
                <c:CardNumber>5210000010001001</c:CardNumber>
                <c:ExpirationDate>2007-07-31</c:ExpirationDate>
            </n:NameCreditCard>
        </n:CreditCards>
        <n:Addresses>
            <n:NameAddress addressType="W" primary="true">
                <c:AddressLine>8512 Yellow Oak St</c:AddressLine>
                <c:CityName>Austin</c:CityName>
                <c:StateProv>TX</c:StateProv>
                <c:CountryCode>US</c:CountryCode>
                <c:PostalCode>78729</c:PostalCode>
            </n:NameAddress>
        </n:Addresses>
        <n:Phones>
            <n:NamePhone phoneType="HOME" phoneRole="PHONE" primary="true">
                <c:PhoneData>
                    <c:PhoneNumber>800-233-0032</c:PhoneNumber>
                </c:PhoneData>
            </n:NamePhone>
        </n:Phones>
    </n:Profile>
  </FetchProfileResponse>
 </soap:Body>
</soap:Envelope>
```

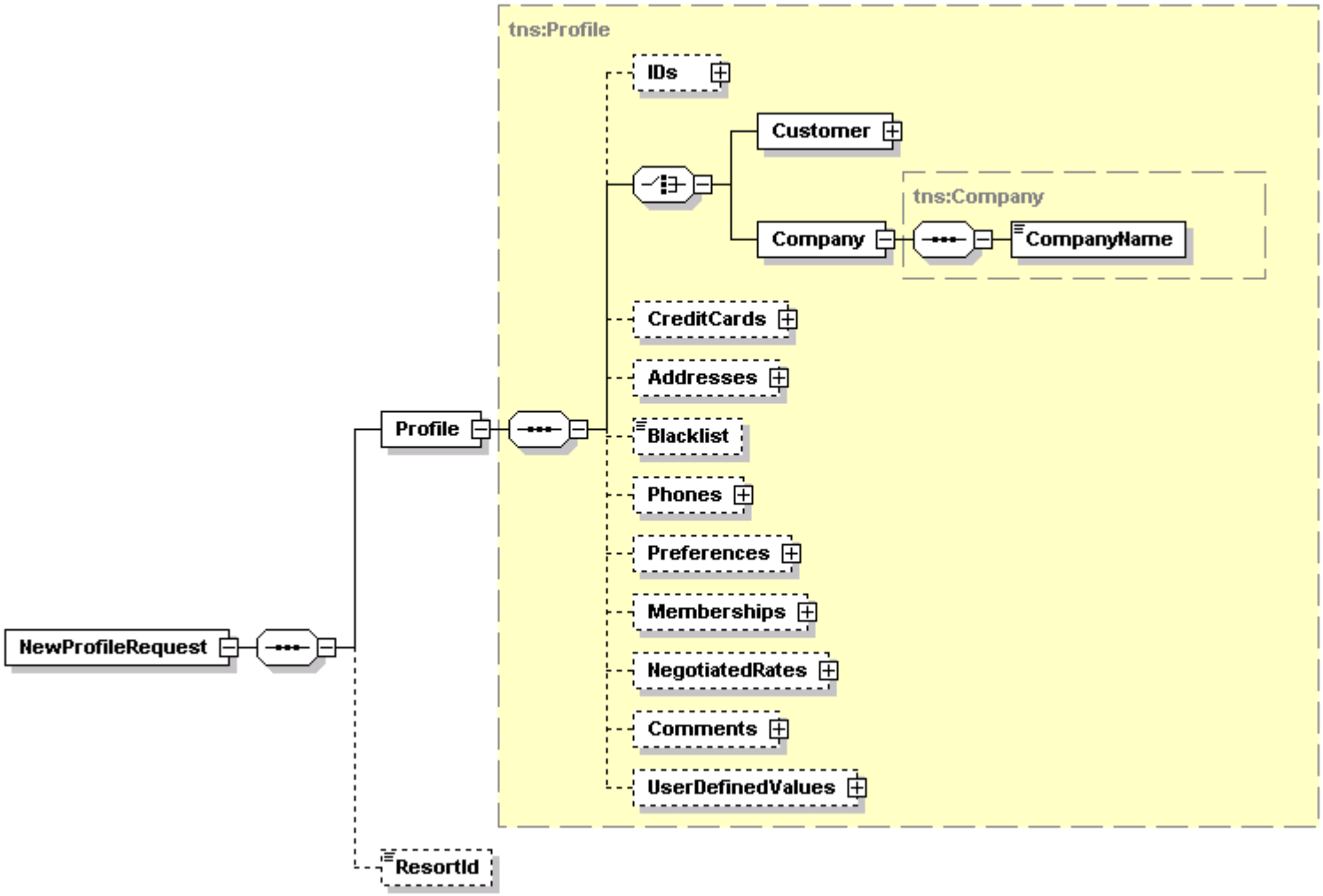

12.2 New Profile

A system may create a new guest profile record in an external system by calling the new profile function.

Port	NamePortType
Binding	NameBinding
Operation	NewProfile

Soap Action	http://htng.org/PWSWG/2006/08/SingleGuestItinerary#NewProfile
Input	NewProfileRequest
Output	NewProfileResponse
Primary Schema	Name.xsd
Role(s) Implemented	Customer Profile System, Activity Reservation System

12.2.1 NewProfileRequest

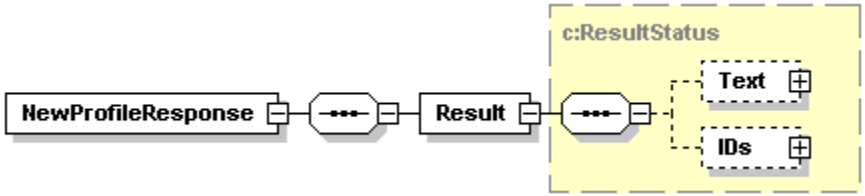


NewProfileRequest
<xs:element name="NewProfileRequest">
<xs:complexType>

```
        <xs:sequence>
            <xs:element name="Profile" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:Profile" />
            <xs:element minOccurs="0" name="ResortId" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
Profile	element	Profile	required	The profile record to create.
ResortId	element	string	optional	The property associated with the profile.

12.2.2 NewProfileResponse



NewProfileResponse

```
<xs:element name="NewProfileResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Result" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:ResultStatus" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
Result	element	ResultStatus	required	The result of the operation.

12.2.3 Sample Message

[*Request*]

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#NewProfile</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:SPASOFT</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:261335f2-bd1d-49e6-96b4-8fcdf2137f8a</wsa:MessageID>
```

```
        <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.micros.com/HTNGActivity/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-d99e21bf-1227-410c-bea3-d8705adcf032">
                <wsu:Created>2006-11-02T20:22:14Z</wsu:Created>
                <wsu:Expires>2008-11-02T20:27:14Z</wsu:Expires>
            </wsu:Timestamp>
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
6f2c0e7f-ac98-4ea3-a726-d67076fc9c72">
                <wsse:Username>HTNG</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">HTNG</wsse:Password>
                <wsse:Nonce>IWGqOAykiUQXnWN5x1wS9Q==</wsse:Nonce>
                <wsu:Created>2006-11-02T20:22:14Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <NewProfileRequest xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types
D:\htng\Name.xsd">
            <Profile nameType="GUEST">
                <IDs>
                    <c:UniqueID source="SPASOFT">567</c:UniqueID>
                </IDs>
                <Customer>
                    <PersonName nameOrdered="String" familiarName="String">
                        <c:LastName>String</c:LastName>
                    </PersonName>
                </Customer>
            </Profile>
            <ResortId>HTNG1</ResortId>
        </NewProfileRequest>
    </soap:Body>
</soap:Envelope>
```


[*Response*]


```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
```

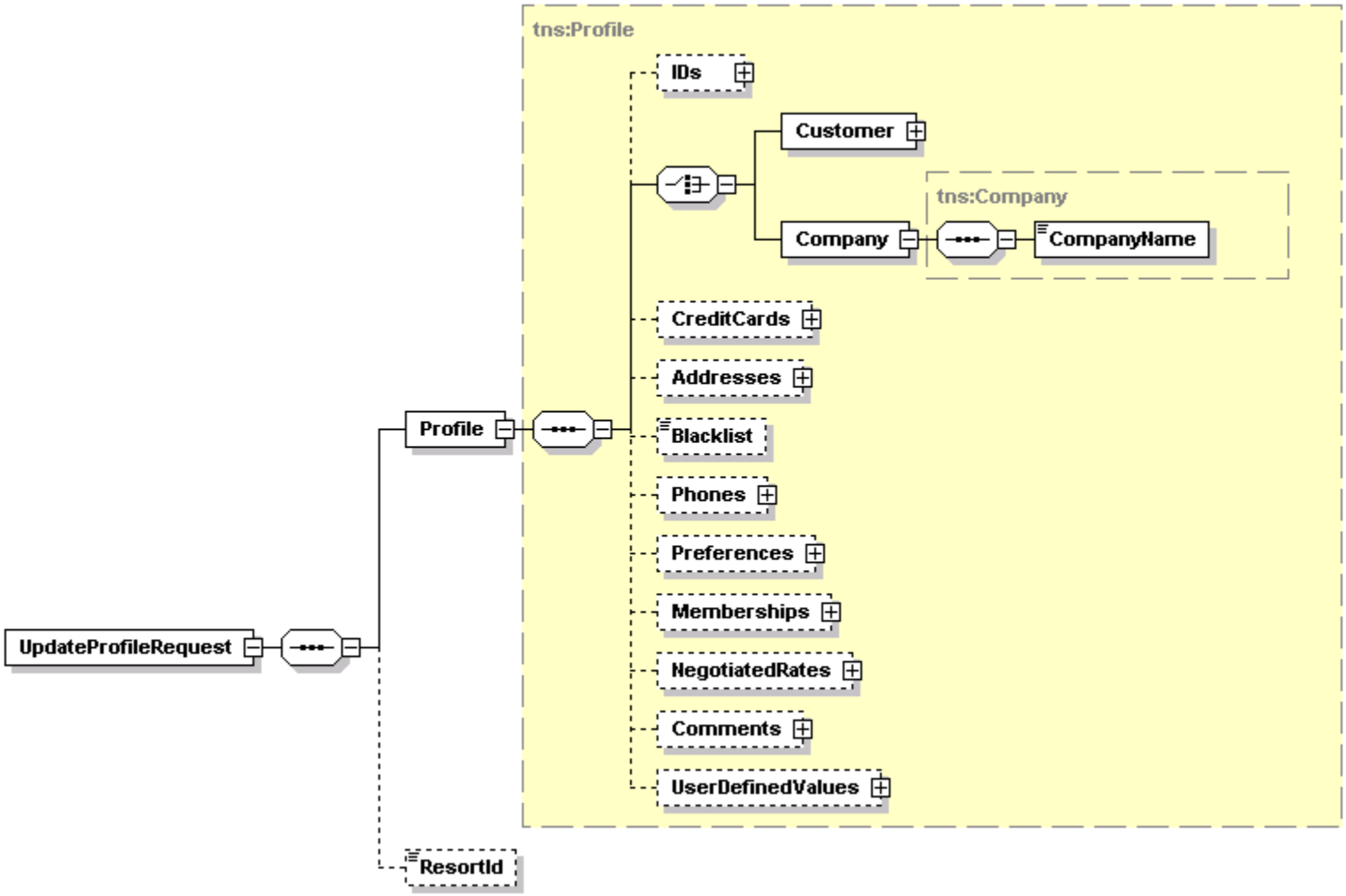
```
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#NewProfileResponse</wsa:Action>
        <wsa:MessageID>urn:uuid:2ef84640-2532-4d42-909b-8c2bfdd13338</wsa:MessageID>
        <wsa:RelatesTo>urn:uuid:261335f2-bd1d-49e6-96b4-8fcdf2137f8a</wsa:RelatesTo>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-deda7b55-7d28-4c77-ae81-575bb4904307">
                <wsu:Created>2006-11-13T21:42:02Z</wsu:Created>
                <wsu:Expires>2006-11-13T21:47:02Z</wsu:Expires>
            </wsu:Timestamp>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <NewProfileResponse xmlns:r="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
xmlns:n="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
xmlns:a="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
xmlns:p="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types">
            <n:Result resultStatusFlag="SUCCESS">
                <c:IDs>
                    <c:UniqueID source="OPERA">2255583</c:UniqueID>
                    <c:UniqueID source="SPASOFT">567</c:UniqueID>
                </c:IDs>
            </n:Result>
        </NewProfileResponse>
    </soap:Body>
</soap:Envelope>
```


12.3 Update Profile

When a linked profile is changed in one system, an update profile message should be sent in order to notify all other systems of the modified record. While there may be a business practice enforced whereby updates are only made in one direction, both systems should be able to accept this message.

Port	NamePortType
Binding	NameBinding
Operation	UpdateProfile
Soap Action	http://htng.org/PWSWG/2006/08/SingleGuestItinerary#UpdateProfile
Input	UpdateProfileRequest
Output	UpdateProfileResponse
Primary Schema	Name.xsd
Role(s) Implemented	Customer Profile System, Activity Reservation System

12.3.1 UpdateProfileRequest

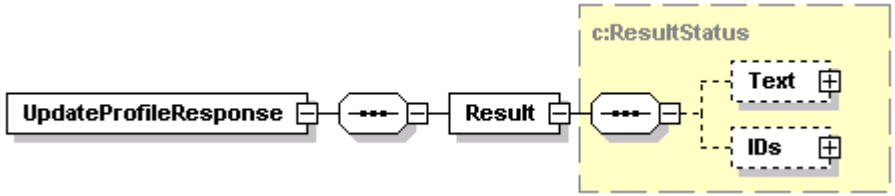


UpdateProfileRequest
<xs:element name="UpdateProfileRequest">   <xs:complexType>

```
        <xs:sequence>
            <xs:element name="Profile" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:Profile" />
            <xs:element minOccurs="0" name="ResortId" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
Profile	element	Profile	required	The profile record to update. This is a full overlay update.
ResortId	element	string	optional	The property the profile is associated with.

12.3.2 UpdateProfileResponse



UpdateProfileResponse

```
<xs:element name="UpdateProfileResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Result" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:ResultStatus" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
Result	element	ResultStatus	required	The result of the update.

12.3.3 Sample Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#UpdateProfile</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:OPERA</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:dcb32613-ee89-44b3-a910-abb165816024</wsa:MessageID>
        <wsa:ReplyTo>
```

```
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.springermiller.com/HTNG_2_0/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-eacdc200-555b-4439-adcb-1c27c2e76291">
                <wsu:Created>2007-02-05T21:30:16Z</wsu:Created>
                <wsu:Expires>2007-02-05T21:35:16Z</wsu:Expires>
            </wsu:Timestamp>
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
3100cb87-5e67-472b-b517-c33c94425569">
                <wsse:Username>OPERA</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">OPERA</wsse:Password>
                <wsse:Nonce>v7hqyEQlgLabyTqlptKmAA==</wsse:Nonce>
                <wsu:Created>2007-02-05T21:30:16Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <UpdateProfileRequest xmlns:r="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
xmlns:n="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
xmlns:a="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
xmlns:p="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types">
            <n:Profile nameType="GUEST">
                <n:IDs>
                    <c:UniqueID source="OPERA">5629459</c:UniqueID>
                </n:IDs>
                <n:Customer gender="UNKNOWN">
                    <n:PersonName familiarName="Sevova">
                        <c:FirstName>Penka</c:FirstName>
                        <c:LastName>Sevova</c:LastName>
                    </n:PersonName>
                </n:Customer>
                <n:Addresses>
                    <n:NameAddress addressType="HOME" primary="true">
                        <c:AddressLine>2640 Us 41 s</c:AddressLine>
                        <c:CityName>Naples</c:CityName>
                        <c:StateProv>FL</c:StateProv>
                        <c:CountryCode>US</c:CountryCode>
                        <c:PostalCode>34119</c:PostalCode>
                    </n:NameAddress>
                    <n:NameAddress addressType="HOME" primary="false">
                        <c:AddressLine>2340 Golden Gate</c:AddressLine>
```

```
                <c:CityName>Naples</c:CityName>
                <c:StateProv>FL</c:StateProv>
                <c:CountryCode>US</c:CountryCode>
                <c:PostalCode>34119</c:PostalCode>
            </n:NameAddress>
            <n:NameAddress addressType="T" primary="false">
                <c:AddressLine>2340 Golden Gate</c:AddressLine>
                <c:CityName>Naples</c:CityName>
                <c:StateProv>FL</c:StateProv>
                <c:CountryCode>US</c:CountryCode>
                <c:PostalCode>34119</c:PostalCode>
            </n:NameAddress>
        </n:Addresses>
        <n:Phones>
            <n:NamePhone phoneType="HOME" phoneRole="PHONE" primary="true">
                <c:PhoneData>
                    <c:PhoneNumber>456</c:PhoneNumber>
                </c:PhoneData>
            </n:NamePhone>
        </n:Phones>
    </n:Profile>
    <n:ResortId>HTNGTEST</n:ResortId>
  </UpdateProfileRequest>
 </soap:Body>
</soap:Envelope>
```


[*Response*]


```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
 <soap:Header>
  <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#UpdateProfileResponse</wsa:Action>
  <wsa:MessageID>urn:uuid:0f166196-b066-4132-bbb0-66ef65f70c1e</wsa:MessageID>
  <wsa:RelatesTo>urn:uuid:dcb32613-ee89-44b3-a910-abb165816024</wsa:RelatesTo>
  <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
  <wsse:Security>
   <wsu:Timestamp wsu:Id="Timestamp-c5614cb2-779c-47fd-8f13-0963177e1ac4">
    <wsu:Created>2007-02-05T21:31:56Z</wsu:Created>
    <wsu:Expires>2007-02-05T21:36:56Z</wsu:Expires>
   </wsu:Timestamp>
  </wsse:Security>
 </soap:Header>
 <soap:Body>
```

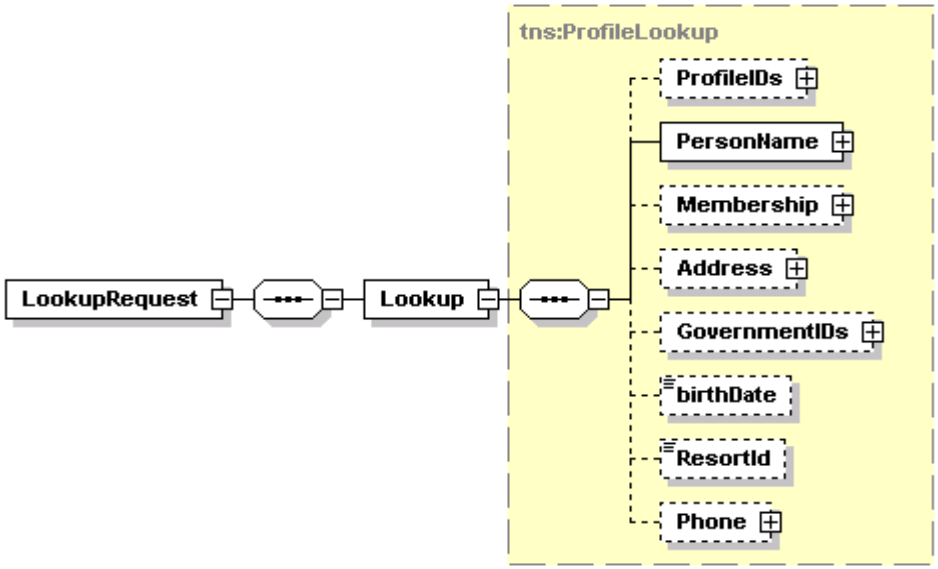
```
        <UpdateProfileResponse xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">
            <Result code="" resultStatusFlag="SUCCESS">
                <c:Text>
                    <c:TextElement language="en"/>
                </c:Text>
                <c:IDs>
                    <c:UniqueID source="OPERA">5629459</c:UniqueID>
                </c:IDs>
            </Result>
        </UpdateProfileResponse>
    </soap:Body>
</soap:Envelope>
```


12.4 Profile Lookup

The profile lookup message may be used in order to request a list of guest records which match a particular set of criteria.

Port	NamePortType
Binding	NameBinding
Operation	ProfileLookup
Soap Action	http://htng.org/PWSWG/2006/08/SingleGuestItinerary#ProfileLookup
Input	ProfileLookupRequest
Output	ProfileLookupResponse
Primary Schema	Name.xsd
Role(s) Implemented	Customer Profile System, Activity Reservation System

12.4.1 LookupRequest



LookupRequest

```
<xs:element name="LookupRequest">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Lookup" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:ProfileLookup" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
Lookup	element	ProfileLookup	required	The lookup filter to apply.

ProfileLookup

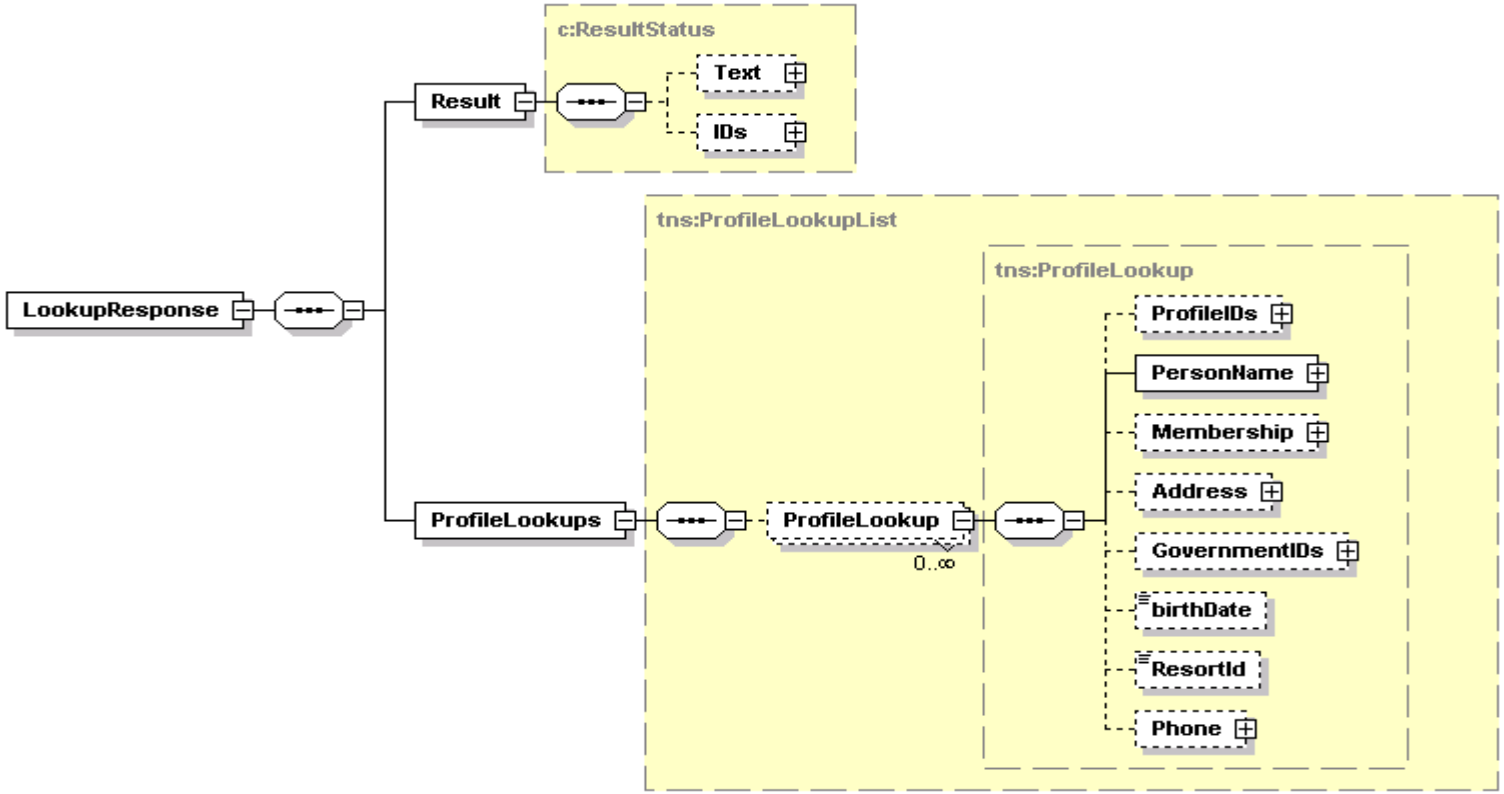
ProfileLookup

```
<xs:complexType name="ProfileLookup">
    <xs:sequence>
        <xs:element minOccurs="0" name="ProfileIDs" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q1:UniqueIDList" />
        <xs:element name="PersonName" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q2:PersonName" />
        <xs:element minOccurs="0" name="Membership" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
```

```
        type="q3:Membership" />
      <xs:element minOccurs="0" name="Address" xmlns:q4="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q4:Address"
        />
      <xs:element minOccurs="0" name="GovernmentIDs" xmlns:q5="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
        type="q5:GovernmentIDList" />
      <xs:element minOccurs="0" name="birthDate" type="xs:date" />
      <xs:element minOccurs="0" name="ResortId" type="xs:string" />
      <xs:element minOccurs="0" name="Phone" xmlns:q6="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q6:Phone" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
ProfileIDs	element	UniqueIDList	optional	A list of ID's to lookup. The receiving system is only required to honor the first value in the list.
PersonName	element	PersonName	required	A filter for an individual's name. The receiving system is only required to lookup by last and first name.
Membership	element	Membership	optional	Membership values to lookup. The receiving system is not required to honor this filter.
Address	element	Address	optional	Address values to lookup. The receiving system can use discretion in applying various components this filter.
GovernmentIDs	element	GovernmentIDList	optional	Government ID lookup values. The receiving system is not required to honor this filter.
birthDate	element	date	optional	Birth date. The receiving system is not required to apply this filter.
ResortId	element	string	optional	The property code.
Phone	element	Phone	optional	A telephone filter. The receiving system is not required to apply this filter.

12.4.2 LookupResponse



LookupResponse

```
<xs:element name="LookupResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Result" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:ResultStatus" />
            <xs:element name="ProfileLookups" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q2:ProfileLookupList"
                />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
Result	element	ResultStatus	required	The result of the operation.

ProfileLookups	element	ProfileLookupList	required	A list of matching profiles.

ProfileLookup

ProfileLookup

```
<xs:complexType name="ProfileLookup">
    <xs:sequence>
        <xs:element minOccurs="0" name="ProfileIDs" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q1:UniqueIDList" />
        <xs:element name="PersonName" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q2:PersonName" />
        <xs:element minOccurs="0" name="Membership" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q3:Membership" />
        <xs:element minOccurs="0" name="Address" xmlns:q4="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q4:Address"
            />
        <xs:element minOccurs="0" name="GovernmentIDs" xmlns:q5="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q5:GovernmentIDList" />
        <xs:element minOccurs="0" name="birthDate" type="xs:date" />
        <xs:element minOccurs="0" name="ResortId" type="xs:string" />
        <xs:element minOccurs="0" name="Phone" xmlns:q6="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q6:Phone" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
ProfileIDs	element	UniqueIDList	optional	The guest record ID.
PersonName	element	PersonName	required	The guest name.
Membership	element	Membership	optional	The guest primary membership record.
Address	element	Address	optional	The guest primary address.
GovernmentIDs	element	GovernmentIDList	optional	A collection of government ID's.
birthDate	element	date	optional	The guest's birth date.
ResortId	element	string	optional	The associated property.
Phone	element	Phone	optional	The guest primary telephone.

12.4.3 Sample Message

[*Request*]

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#ProfileLookup</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:SPASOFT</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:4ef62b8b-72dc-4125-a403-dca797465a20</wsa:MessageID>
```

```
        <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.micros.com/HTNGActivity/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-42887188-d0d4-4297-92e5-8091b18c963a">
                <wsu:Created>2006-11-02T20:18:21Z</wsu:Created>
                <wsu:Expires>2006-11-02T20:23:21Z</wsu:Expires>
            </wsu:Timestamp>
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
b5059290-0922-4492-845d-e4c7148a2a6d">
                <wsse:Username>HTNG</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">HTNG</wsse:Password>
                <wsse:Nonce>Fnacfly4VV9e1XSujzrgvQ==</wsse:Nonce>
                <wsu:Created>2006-11-02T20:18:21Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <LookupRequest xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">
            <Lookup>
                <ProfileIDs>
                    <c:UniqueID source=""/>
                </ProfileIDs>
                <PersonName nameOrdered="" familiarName="">
                    <NameTitle xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"/>
                    <FirstName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"/>
                    <MiddleName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"/>
                    <LastName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">Brady</LastName>
                    <NameSuffix xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"/>
                </PersonName>
                <Address otherAddressType="" addressType="">
                    <AddressLine xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"/>
                    <CityName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"/>
                    <StateProv xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"/>
                    <CountryCode xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"/>
                    <PostalCode xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"/>
                </Address>
                <ResortId>HTNG1</ResortId>
                <Phone phoneRole="" phoneType="">
                    <PhoneNumber xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"/>
                </Phone>
            </Lookup>
```

```
        </LookupRequest>
    </soap:Body>
</soap:Envelope>
```


[*Response*]


```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#ProfileLookupResponse</wsa:Action>
        <wsa:MessageID>urn:uuid:1cdb442c-bab2-4e37-852b-8e85cd2693b9</wsa:MessageID>
        <wsa:RelatesTo>urn:uuid:4ef62b8b-72dc-4125-a403-dca797465a20</wsa:RelatesTo>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-46f645d8-b426-4984-9c2e-6c60a8a7da47">
                <wsu:Created>2006-11-02T20:17:25Z</wsu:Created>
                <wsu:Expires>2006-11-02T20:22:25Z</wsu:Expires>
            </wsu:Timestamp>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <LookupResponse xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types">
            <Result resultStatusFlag="SUCCESS"/>
            <ProfileLookups>
                <ProfileLookup>
                    <ProfileIDs>
                        <UniqueID source="OPERA" xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">168609</UniqueID>
                    </ProfileIDs>
                    <PersonName>
                        <FirstName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">MICHAEL F</FirstName>
                        <LastName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">BRADY</LastName>
                    </PersonName>
                    <Address/>
                </ProfileLookup>
                <ProfileLookup>
                    <ProfileIDs>
                        <UniqueID source="OPERA" xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">171294</UniqueID>
                    </ProfileIDs>
                    <PersonName>
                        <FirstName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">DON</FirstName>
                        <LastName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">BRADY</LastName>
                    </PersonName>
                    <Address/>
```

```
</ProfileLookup>
<ProfileLookup>
    <ProfileIDs>
        <UniqueID source="OPERA" xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">172019</UniqueID>
    </ProfileIDs>
    <PersonName>
        <FirstName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">ROBERT E</FirstName>
        <LastName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">BRADY</LastName>
    </PersonName>
    <Address/>
</ProfileLookup>
<ProfileLookup>
    <ProfileIDs>
        <UniqueID source="OPERA" xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">383254</UniqueID>
    </ProfileIDs>
    <PersonName>
        <FirstName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">RAY</FirstName>
        <LastName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">BRADY</LastName>
    </PersonName>
    <Address>
        <AddressLine xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">440 FAIR DR</AddressLine>
        <CityName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">COSTA MESA</CityName>
        <StateProv xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">CA</StateProv>
        <CountryCode xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">US</CountryCode>
        <PostalCode xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">92626</PostalCode>
    </Address>
</ProfileLookup>
<ProfileLookup>
    <ProfileIDs>
        <UniqueID source="OPERA" xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">406967</UniqueID>
    </ProfileIDs>
    <PersonName>
        <FirstName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">MATT</FirstName>
        <LastName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">BRADY</LastName>
    </PersonName>
    <Address/>
</ProfileLookup>
<ProfileLookup>
    <ProfileIDs>
        <UniqueID source="OPERA" xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">406244</UniqueID>
    </ProfileIDs>
    <PersonName>
        <FirstName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">LAURIE M</FirstName>
        <LastName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">BRADY</LastName>
    </PersonName>
```

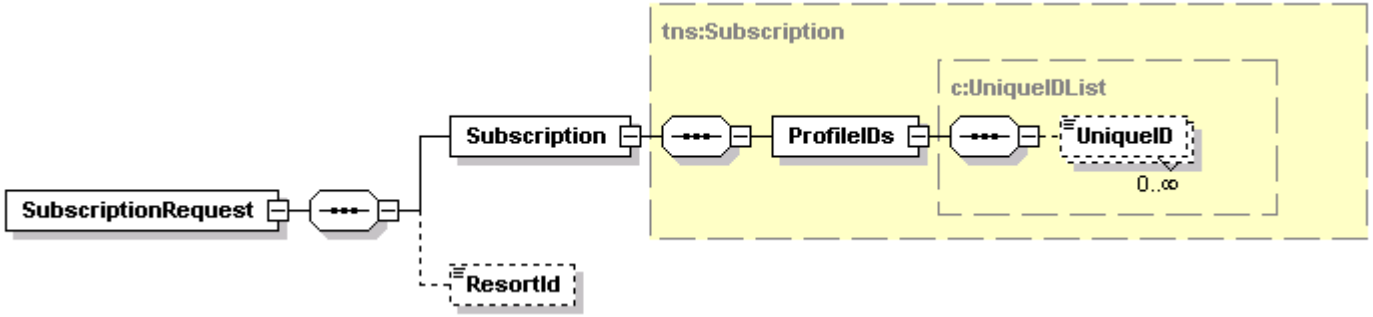
```
            <Address/>
        </ProfileLookup>
        <ProfileLookup>
            <ProfileIDs>
                <UniqueID source="OPERA" xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">413792</UniqueID>
            </ProfileIDs>
            <PersonName>
                <FirstName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">VICTORIA</FirstName>
                <LastName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">BRADY</LastName>
            </PersonName>
            <Address>
                <AddressLine xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">40 PAGET RD.</AddressLine>
                <CityName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">MADISON</CityName>
                <StateProv xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">WI.</StateProv>
                <CountryCode xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">US</CountryCode>
                <PostalCode xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">53704</PostalCode>
            </Address>
        </ProfileLookup>
    </ProfileLookups>
  </LookupResponse>
</soap:Body>
</soap:Envelope>
```


12.5 Subscription

A subscription message is sent to notify an external system after a new profile has been created, of the internal ID for the new record. The typical scenario consists of a sequence of three actions: A profile lookup, a profile fetch, and a subscription. Often, when a profile is fetched from an external system, it does not exist in the requesting system and must be created. The subscription record is a notification back to the other system of the record ID for the newly created guest record. While the requesting system would have both ID's, the external system would otherwise not be aware of the profile link.

Port	NamePortType
Binding	NameBinding
Operation	Subscription
Soap Action	http://htng.org/PWSWG/2006/08/SingleGuestItinerary#Subscription
Input	SubscriptionRequest
Output	SubscriptionResponse
Primary Schema	Name.xsd
Role(s) Implemented	Customer Profile System, Activity Reservation System

12.5.1 SubscriptionRequest

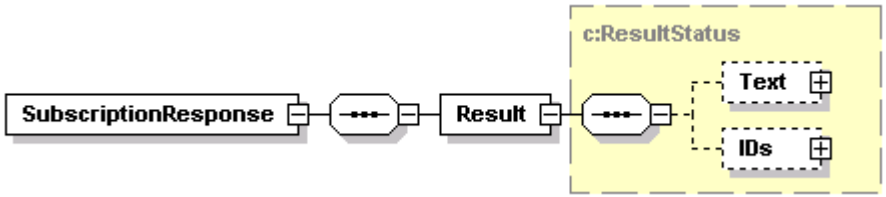


SubscriptionRequest				
<xs:element name="SubscriptionRequest">
<xs:complexType>
<xs:sequence>
<xs:element name="Subscription" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:Subscription" />
<xs:element minOccurs="0" name="ResortId" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>				
Name	**Type**	**Data Type**	**Use**	**Comments**
Subscription	element	Subscription	required	A matched pair of external and internal ID values.
ResortId	element	string	optional	The associated property code.

Subscription

Subscription				
<xs:complexType name="Subscription">
<xs:sequence>
<xs:element name="ProfileIDs" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueIDList" />
</xs:sequence>
<xs:attribute name="subscriptionAction" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q2:SubscriptionAction" use="required" />
<xs:attribute name="otherSubscriptionAction" type="xs:string" />
</xs:complexType>				
Name	**Type**	**Data Type**	**Use**	**Comments**
subscriptionAction	attribute	SubscriptionAction	required	One of *OTHER*, *SUBSCRIBE*, or *UNSUBSCRIBE*.
otherSubscriptionAction	attribute	string	none	Not used.
ProfileIDs	element	UniqueIDList	required	A matched pair of internal and external guest record ID's.

12.5.2 SubscriptionResponse



SubscriptionResponse				
<xs:element name="SubscriptionResponse">
<xs:complexType>
<xs:sequence>
<xs:element name="Result" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:ResultStatus" />
</xs:sequence>
</xs:complexType>
</xs:element>				
Name	**Type**	**Data Type**	**Use**	**Comments**
Result	element	ResultStatus	required	The result of the subscription.

12.5.3 Sample Message

[*Request*]

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#Subscription</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:SPASOFT</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:5518896e-eed9-4a8b-89b6-4c4a5a3f756f</wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.micros.com/HTNGActivity/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-aeadcfcd-62bb-4569-9adf-2058cfd1f130">
                <wsu:Created>2006-11-08T17:17:45Z</wsu:Created>
                <wsu:Expires>2006-11-08T17:22:45Z</wsu:Expires>
            </wsu:Timestamp>
```

```
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
3bdd620a-3146-4012-8da3-a202e5b1f896">
                <wsse:Username>HTNG</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">HTNG</wsse:Password>
                <wsse:Nonce>rtQ+bYWTWfGrRipi6fhngg==</wsse:Nonce>
                <wsu:Created>2006-11-08T17:17:45Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <SubscriptionRequest xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">
            <Subscription otherSubscriptionAction="" subscriptionAction="SUBSCRIBE">
                <ProfileIDs>
                    <c:UniqueID source="OPERA">188457</c:UniqueID>
                    <c:UniqueID source="SPASOFT">519</c:UniqueID>
                </ProfileIDs>
            </Subscription>
            <ResortId>HTNG1</ResortId>
        </SubscriptionRequest>
    </soap:Body>
</soap:Envelope>
```


[*Response*]


```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#SubscriptionResponse</wsa:Action>
        <wsa:MessageID>urn:uuid:0fe8b10d-2837-40ab-8d8d-2b11c80fb92b</wsa:MessageID>
        <wsa:RelatesTo>urn:uuid:5518896e-eed9-4a8b-89b6-4c4a5a3f756f</wsa:RelatesTo>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-cde37d00-4c6f-4a48-ba04-67fd2b523f5e">
                <wsu:Created>2006-11-08T17:19:07Z</wsu:Created>
                <wsu:Expires>2006-11-08T17:24:07Z</wsu:Expires>
            </wsu:Timestamp>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <SubscriptionResponse xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types">
```

```
            <Result resultStatusFlag="SUCCESS"/>
        </SubscriptionResponse>
    </soap:Body>
</soap:Envelope>
```

Chapter 13 Reservation Provider Port Type

The Reservation Provider Port consists of four functions related to a guest reservation in an Accommodation Reservation System. The functions provide the means to lookup a reservation based on various filters, fetch reservation details, and post payments to a reservation.

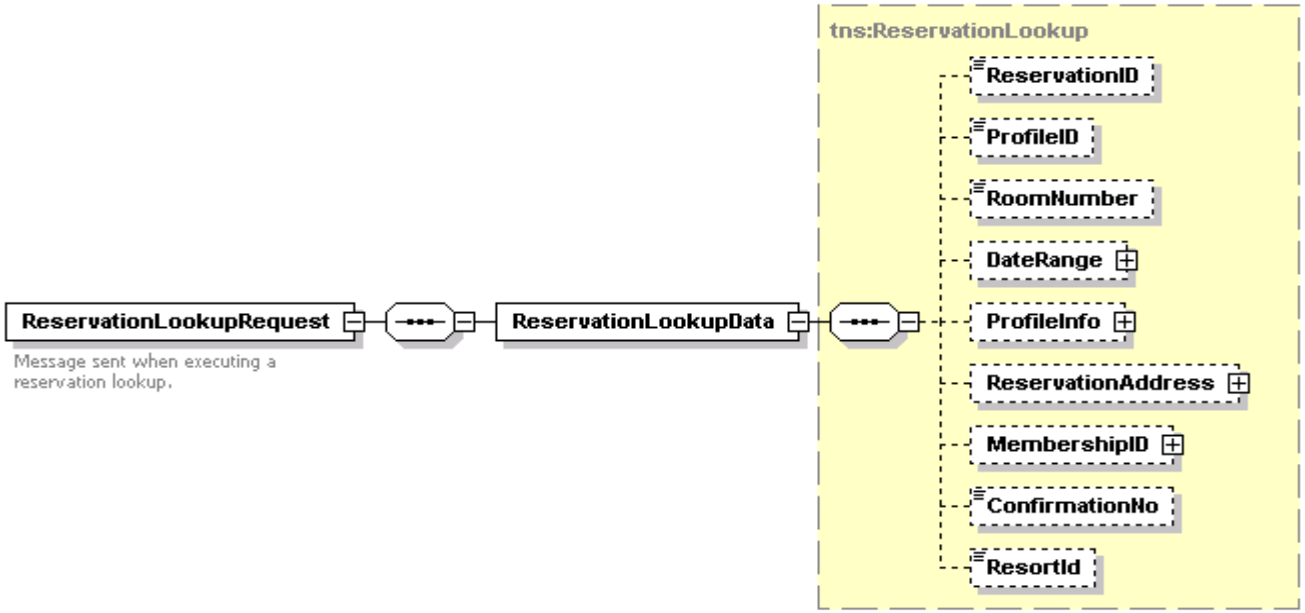
- ReservationLookup
  - Request a list of guest reservations with match one or more input filters.
- FetchReservation
  - Fetch details for a guest reservation.
- PostPayment
  - Post a payment to a guest reservation.
- GuestMessage
  - Submit a message to be delivered to a guest and provide notification.

13.1 Reservation Lookup

The Accommodation Reservation System provides the reservation lookup message in order to request a list of guest reservations matching a specified set of criteria. This is often used by the Activity Reservation System to look up a guest record by room number or confirmation number.

Port	ReservationProviderPortType
Binding	ReservationProviderBinding
Operation	ReservationLookup
Soap Action	http://htng.org/PWSWG/2006/08/SingleGuestItinerary#ReservationLookup
Input	ReservationLookupRequest
Output	ReservationLookupResponse
Primary Schema	Reservation.xsd
Role(s) Implemented	Accommodation Reservation System

13.1.1 ReservationLookupRequest



ReservationLookupRequest

```
<xs:element name="ReservationLookupRequest">
    <xs:annotation>
    <xs:documentation>Message sent when executing a reservation lookup.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ReservationLookupData" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
                type="q1:ReservationLookup" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
ReservationLookupData	element	ReservationLookup	required	Reservation lookup filter data.

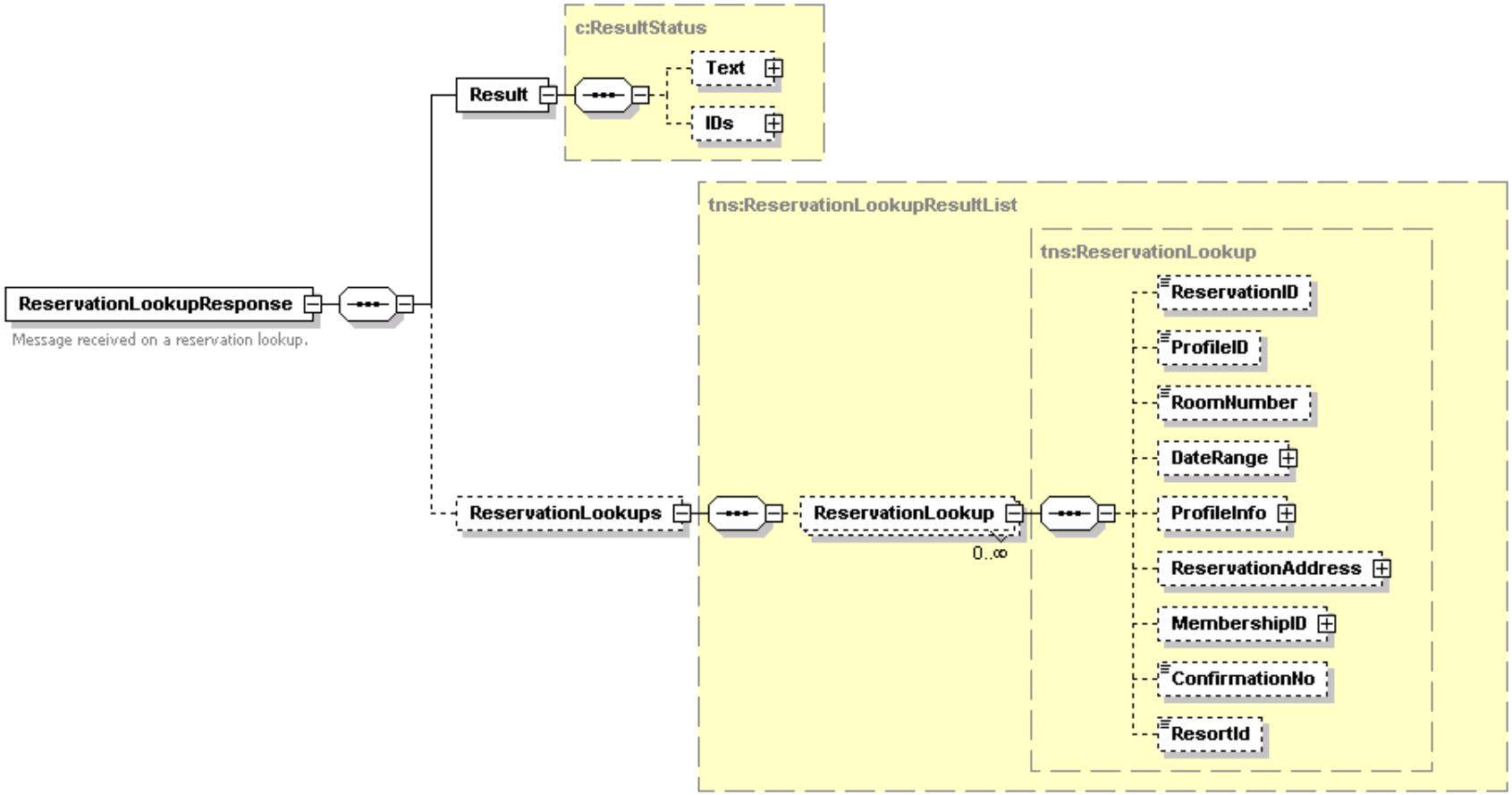
ReservationLookup

ReservationLookup

```
<xs:complexType name="ReservationLookup">
    <xs:annotation>
    <xs:documentation>Defines the queryied data when executing a reservation lookup.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element minOccurs="0" name="ReservationID" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q1:UniqueID" />
        <xs:element minOccurs="0" name="ProfileID" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q2:UniqueID"
            />
        <xs:element minOccurs="0" name="RoomNumber" type="xs:string" />
        <xs:element minOccurs="0" name="DateRange" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
            type="q3:OGTimeSpan" />
        <xs:element minOccurs="0" name="ProfileInfo" xmlns:q4="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q4:PersonName" />
        <xs:element minOccurs="0" name="ReservationAddress" xmlns:q5="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q5:Address" />
        <xs:element minOccurs="0" name="MembershipID" xmlns:q6="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q6:Membership" />
        <xs:element minOccurs="0" name="ConfirmationNo" type="xs:string" />
        <xs:element minOccurs="0" name="ResortId" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="reservationStatus" xmlns:q7="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
        type="q7:ReservationStatusType" use="required" />
    <xs:attribute name="otherReservationStatus" type="xs:string" use="optional" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
reservationStatus	attribute	ReservationStatusType	required	Status of reservation.
otherReservationStatus	attribute	string	optional	Not used.
ReservationID	element	UniqueID	optional	Lookup by reservation ID.
ProfileID	element	UniqueID	optional	Lookup by profile ID.
RoomNumber	element	string	optional	Lookup by room number.
DateRange	element	OGTimeSpan	optional	Lookup by date range.
ProfileInfo	element	PersonName	optional	Lookup by guest name.
ReservationAddress	element	Address	optional	Lookup by guest address.
MembershipID	element	Membership	optional	Lookup by guest membership.
ConfirmationNo	element	string	optional	Lookup by confirmation number.
ResortId	element	string	optional	Property code to lookup from.

13.1.2 ReservationLookupResponse



ReservationLookupResponse
<xs:element name="ReservationLookupResponse"> <xs:annotation> <xs:documentation>Message received on a reservation lookup.</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence>

```
        <xs:element name="Result" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:ResultStatus" />
        <xs:element minOccurs="0" name="ReservationLookups" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
            type="q2:ReservationLookupResultList" />
      </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
Result	element	ResultStatus	required	Result of the lookup operation.
ReservationLookups	element	ReservationLookupResultList	optional	Collection of matching reservation summary records.

ReservationLookupResultList

ReservationLookupResultList

```
<xs:complexType name="ReservationLookupResultList">
    <xs:annotation>
    <xs:documentation>Collection of reservation lookup results.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="ReservationLookup"
          xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types" type="q1:ReservationLookup" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
ReservationLookup	element	ReservationLookup	optional / multiple	Collection of ReservationLookup elements.

ReservationLookup

ReservationLookup

```
<xs:complexType name="ReservationLookup">
    <xs:annotation>
    <xs:documentation>Defines the queryied data when executing a reservation lookup.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element minOccurs="0" name="ReservationID" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
          type="q1:UniqueID" />
      <xs:element minOccurs="0" name="ProfileID" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q2:UniqueID"
          />
      <xs:element minOccurs="0" name="RoomNumber" type="xs:string" />
      <xs:element minOccurs="0" name="DateRange" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
          type="q3:OGTimeSpan" />
      <xs:element minOccurs="0" name="ProfileInfo" xmlns:q4="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
          type="q4:PersonName" />
      <xs:element minOccurs="0" name="ReservationAddress" xmlns:q5="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
```

```
            type="q5:Address" />
        <xs:element minOccurs="0" name="MembershipID" xmlns:q6="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q6:Membership" />
        <xs:element minOccurs="0" name="ConfirmationNo" type="xs:string" />
        <xs:element minOccurs="0" name="ResortId" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="reservationStatus" xmlns:q7="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
        type="q7:ReservationStatusType" use="required" />
    <xs:attribute name="otherReservationStatus" type="xs:string" use="optional" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
reservationStatus	attribute	ReservationStatusType	required	The reservation status.
otherReservationStatus	attribute	string	optional	Not used.
ReservationID	element	UniqueID	optional	The internal reservation ID from the message receiver.
ProfileID	element	UniqueID	optional	The internal guest record ID from themessage receiver.
RoomNumber	element	string	optional	The guest room number.
DateRange	element	OGTimeSpan	optional	The guest arrival and departure dates.
ProfileInfo	element	PersonName	optional	The guest name.
ReservationAddress	element	Address	optional	The guest primary address.
MembershipID	element	Membership	optional	The guest primary membership.
ConfirmationNo	element	string	optional	The reservation confirmation number.
ResortId	element	string	optional	The property code.

13.1.3 Sample Message

[*Request*]

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#ReservationLookup</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:SPASOFT</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:393e560b-ea97-4c8f-b5f7-16bc165c4bbc</wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.micros.com/HTNGActivity/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-41a9938a-d6d9-48a6-913f-ac1be51d663a">
                <wsu:Created>2006-11-02T14:48:24Z</wsu:Created>
                <wsu:Expires>2006-11-02T14:53:24Z</wsu:Expires>
```

```
            </wsu:Timestamp>
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
b2ebe37c-c028-457b-8fbf-45896fdea13d">
                <wsse:Username>HTNG</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">HTNG</wsse:Password>
                <wsse:Nonce>zqlSwFYwV50GNgVL4Fl/uQ==</wsse:Nonce>
                <wsu:Created>2006-11-02T14:48:24Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <ReservationLookupRequest xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
xmlns:a="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">
            <ReservationLookupData>
                <DateRange>
                    <Start xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types">2006-11-02T00:00:00</Start>
                    <End xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types">2006-11-05T00:00:00</End>
                </DateRange>
                <ProfileInfo>
                    <FirstName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">Vernon</FirstName>
                    <LastName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">Smith</LastName>
                </ProfileInfo>
                <ResortId>HTNG1</ResortId>
            </ReservationLookupData>
        </ReservationLookupRequest>
    </soap:Body>
</soap:Envelope>
```


[*Response*]


```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#ReservationLookupResponse</wsa:Action>
        <wsa:MessageID>urn:uuid:8457a488-2db0-448e-919f-b0e94dcef734</wsa:MessageID>
        <wsa:RelatesTo>urn:uuid:393e560b-ea97-4c8f-b5f7-16bc165c4bbc</wsa:RelatesTo>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-16b03c8a-7959-46e7-a23c-d3333c5d9170">
                <wsu:Created>2006-11-02T14:47:21Z</wsu:Created>
                <wsu:Expires>2006-11-02T14:52:21Z</wsu:Expires>
```

```
            </wsu:Timestamp>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <ReservationLookupResponse xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types">
            <Result resultStatusFlag="SUCCESS"/>
            <ReservationLookups>
                <ReservationLookup reservationStatus="CHECKED_IN">
                    <ReservationID source="OPERA">1620979</ReservationID>
                    <ProfileID source="OPERA">2254583</ProfileID>
                    <RoomNumber>10</RoomNumber>
                    <DateRange>
                        <Start xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types">2006-11-02T00:00:00</Start>
                        <End xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types">2006-11-05T00:00:00</End>
                    </DateRange>
                    <ProfileInfo>
                        <FirstName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">Vernon</FirstName>
                        <LastName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">Smith</LastName>
                    </ProfileInfo>
                    <ReservationAddress>
                        <AddressLine xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">Building 2</AddressLine>
                        <AddressLine xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">1234 Aligator way</AddressLine>
                        <CityName xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">Vanderbilt Beach</CityName>
                        <StateProv xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">FL</StateProv>
                        <CountryCode xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">US</CountryCode>
                        <PostalCode xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">34107</PostalCode>
                    </ReservationAddress>
                    <MembershipID/>
                    <ConfirmationNo>2970622</ConfirmationNo>
                    <ResortId>HTNG1</ResortId>
                </ReservationLookup>
            </ReservationLookups>
        </ReservationLookupResponse>
    </soap:Body>
</soap:Envelope>
```

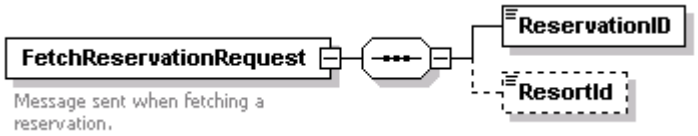

13.2 Fetch Reservation

The fetch reservation message is used to request details for a particular reservation based on an associated ID defined within the receiving Accommodation Reservation System. In many instances, this request is made following a reservation lookup where one reservation is selected from the list provided.

Port	ReservationProviderPortType
Binding	ReservationProviderBinding
Operation	ReservationLookup
Soap Action	http://htng.org/PWSWG/2006/08/SingleGuestItinerary#FetchReservation
Input	ReservationLookupRequest

Output	ReservationLookupResponse
Primary Schema	Reservation.xsd
Role(s) Implemented	Accommodation Reservation System

13.2.1 FetchReservationRequest

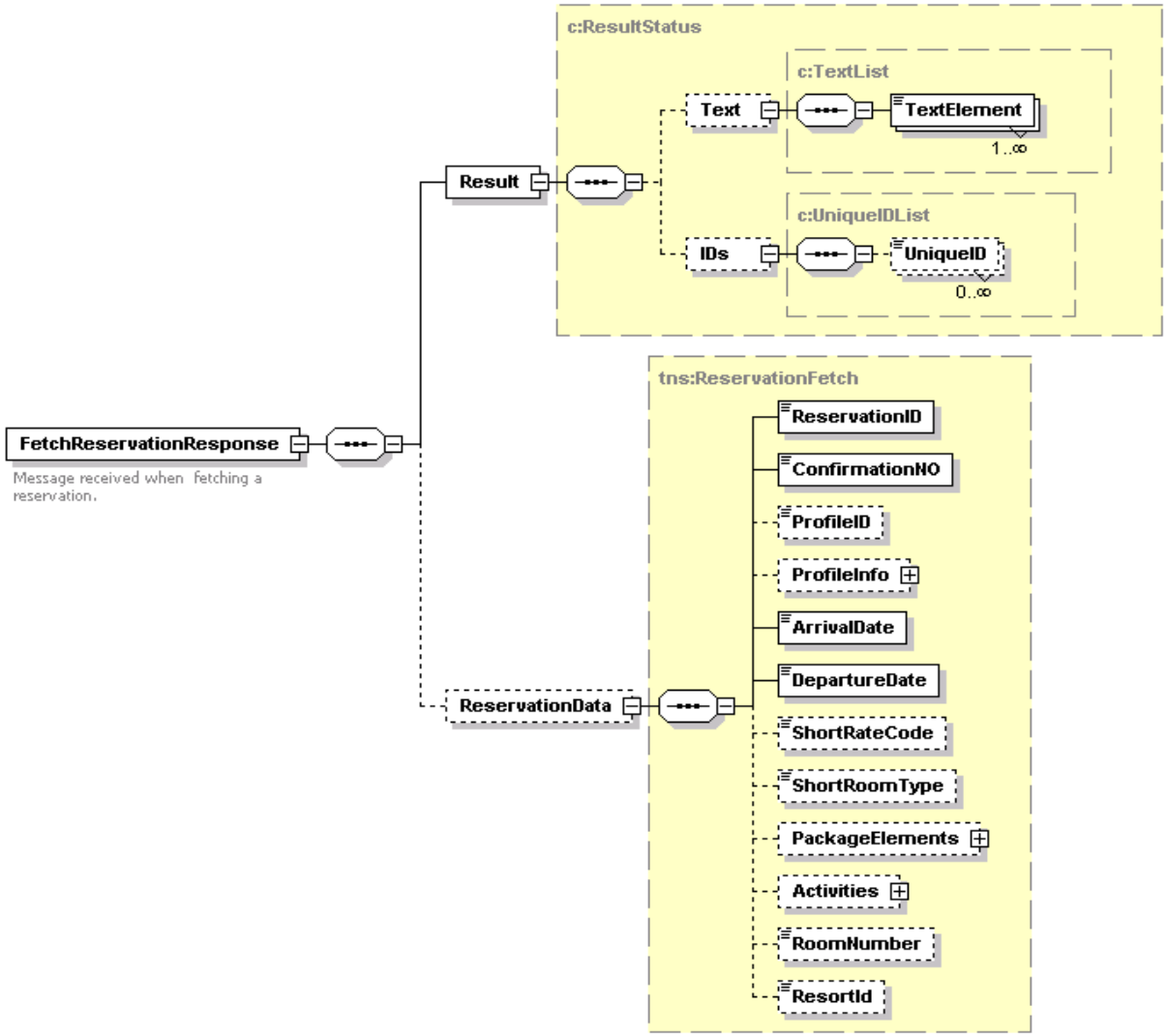


FetchReservationRequest

```
<xs:element name="FetchReservationRequest">
    <xs:annotation>
    <xs:documentation>Message sent when fetching a reservation.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ReservationID" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueID" />
            <xs:element minOccurs="0" name="ResortId" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
ReservationID	element	UniqueID	required	The reservation ID to fetch.
ResortId	element	string	optional	The property code.

13.2.2 FetchReservationResponse



FetchReservationResponse
<xs:element name="FetchReservationResponse">

```
    <xs:annotation>
    <xs:documentation>Message received when  fetching a reservation.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Result" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:ResultStatus" />
            <xs:element minOccurs="0" name="ReservationData" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
                type="q2:ReservationFetch" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
Result	element	ResultStatus	required	The result of the operation.
ReservationData	element	ReservationFetch	optional	Reservation details.

ReservationFetch

ReservationFetch

```
<xs:complexType name="ReservationFetch">
    <xs:annotation>
    <xs:documentation>Data returned when fetching a reservation.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="ReservationID" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueID" />
        <xs:element name="ConfirmationNO" type="xs:string" />
        <xs:element minOccurs="0" name="ProfileID" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q2:UniqueID"
            />
        <xs:element minOccurs="0" name="ProfileInfo" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q3:PersonName" />
        <xs:element name="ArrivalDate" type="xs:dateTime" />
        <xs:element name="DepartureDate" type="xs:dateTime" />
        <xs:element minOccurs="0" name="ShortRateCode" type="xs:string" />
        <xs:element minOccurs="0" name="ShortRoomType" type="xs:string" />
        <xs:element minOccurs="0" name="PackageElements" xmlns:q4="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
            type="q4:PackageElementList" />
        <xs:element minOccurs="0" name="Activities" xmlns:q5="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types" type="q5:ActivityList"
            />
        <xs:element minOccurs="0" name="RoomNumber" type="xs:string" />
        <xs:element minOccurs="0" name="ResortId" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="reservationStatus" xmlns:q6="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
        type="q6:ReservationStatusType" use="required" />
    <xs:attribute name="otherReservationStatus" type="xs:string" use="optional" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments

reservationStatus	attribute	ReservationStatusType	required	The reservation status.
otherReservationStatus	attribute	string	optional	Not used.
ReservationID	element	UniqueID	required	The internal reservation ID.
ConfirmationNO	element	string	required	The confirmation number.
ProfileID	element	UniqueID	optional	The primary guest profile ID.
ProfileInfo	element	PersonName	optional	The primary guest name.
ArrivalDate	element	dateTime	required	The arrival date.
DepartureDate	element	dateTime	required	The departure date.
ShortRateCode	element	string	optional	The reservatuib rate code.
ShortRoomType	element	string	optional	The room type description.
PackageElements	element	PackageElementList	optional	A collection of package elements.
Activities	element	ActivityList	optional	A collection of activities.
RoomNumber	element	string	optional	The room number.
ResortId	element	string	optional	The property code.

PackageElementList

PackageElementList

```
<xs:complexType name="PackageElementList">
    <xs:annotation>
    <xs:documentation>A collection of packages attached to a reservation.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element maxOccurs="unbounded" name="PackageElement" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
            type="q1:PackageElement" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
PackageElement	element	PackageElement	required / multiple	A collection of PackageElement records.

PackageElement

PackageElement

```
<xs:complexType name="PackageElement">
    <xs:annotation>
    <xs:documentation>Package data attached to a reservation.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="ElementCode" type="xs:string" />
        <xs:element name="ElementDescription" type="xs:string" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments

ElementCode	element	string	required	Package element code.
ElementDescription	element	string	required	Package element description.

13.2.3 Sample Message

[*Request*]

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#FetchReservation</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:SPASOFT</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:261335f2-bd1d-49e6-96b4-8fcdf2137f8a</wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.micros.com/HTNGActivity/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-d99e21bf-1227-410c-bea3-d8705adcf032">
                <wsu:Created>2006-11-02T20:22:14Z</wsu:Created>
                <wsu:Expires>2008-11-02T20:27:14Z</wsu:Expires>
            </wsu:Timestamp>
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
6f2c0e7f-ac98-4ea3-a726-d67076fc9c72">
                <wsse:Username>HTNG</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">HTNG</wsse:Password>
                <wsse:Nonce>IWGqOAykiUQXnWN5x1wS9Q==</wsse:Nonce>
                <wsu:Created>2006-11-02T20:22:14Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <FetchReservationRequest xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
xmlns:a="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types
D:\htng\Reservation.xsd">
            <ReservationID source="OPERA">1621229</ReservationID>
            <ResortId>HTNG1</ResortId>
        </FetchReservationRequest>
    </soap:Body>
```

```
</soap:Envelope>
```

[*Response*]

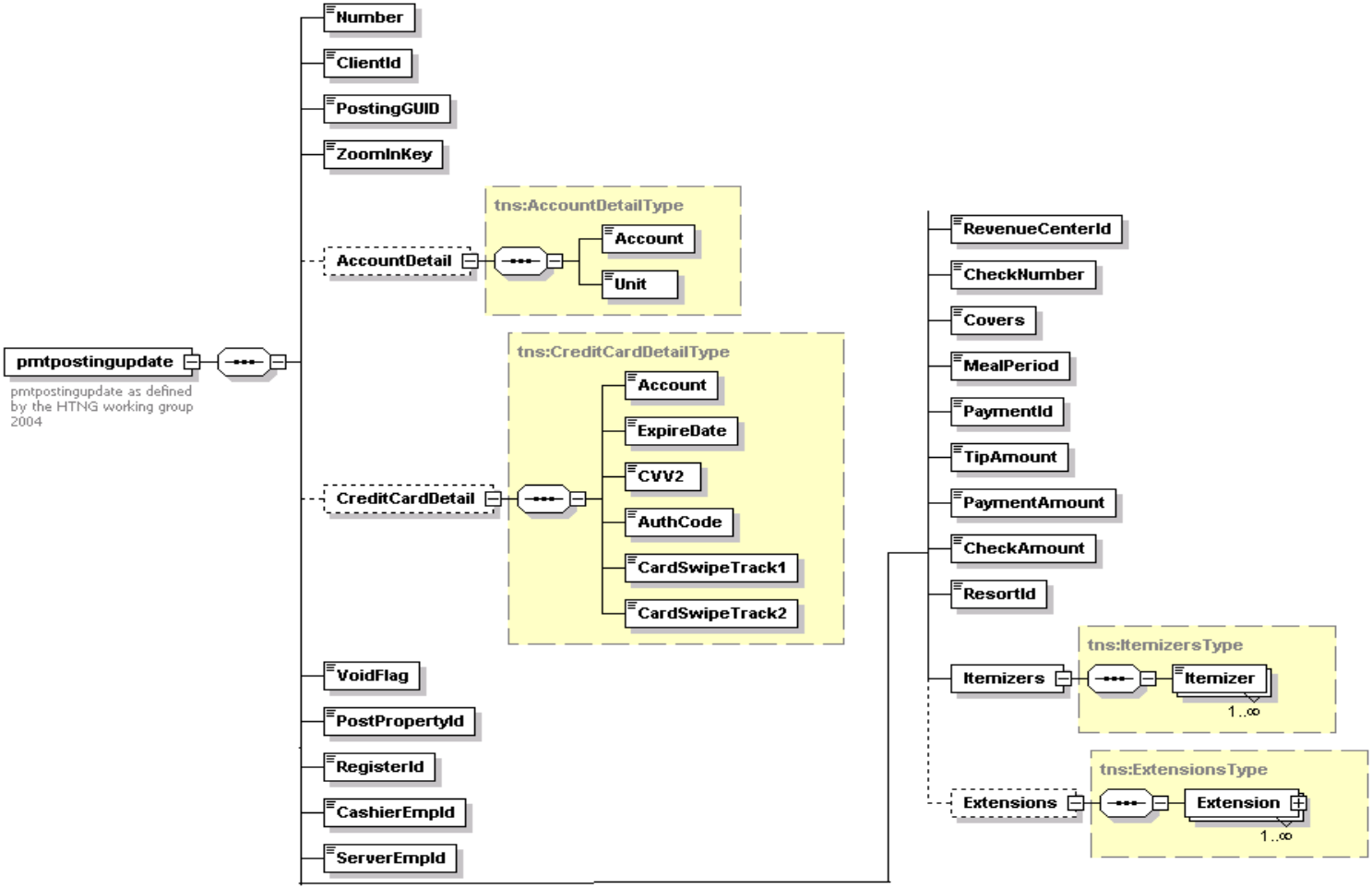

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#FetchReservationResponse</wsa:Action>
        <wsa:MessageID>urn:uuid:ff184ca9-18c0-4ce0-bab1-4b6dbb394135</wsa:MessageID>
        <wsa:RelatesTo>urn:uuid:261335f2-bd1d-49e6-96b4-8fcdf2137f8a</wsa:RelatesTo>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-cd3c3098-f9c9-4c35-bb1d-a3e4b020e8f6">
                <wsu:Created>2006-11-14T13:13:45Z</wsu:Created>
                <wsu:Expires>2006-11-14T13:18:45Z</wsu:Expires>
            </wsu:Timestamp>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <FetchReservationResponse xmlns:r="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
xmlns:n="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
xmlns:a="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
xmlns:p="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types">
            <r:Result resultStatusFlag="SUCCESS"/>
            <r:ReservationData reservationStatus="RESERVED">
                <r:ReservationID source="OPERA">1621229</r:ReservationID>
                <r:ConfirmationNO>2970872</r:ConfirmationNO>
                <r:ProfileID source="OPERA">450018</r:ProfileID>
                <r:ProfileInfo>
                    <c:FirstName>TIM</c:FirstName>
                    <c:LastName>VANCE</c:LastName>
                </r:ProfileInfo>
                <r:ArrivalDate>2006-12-01T00:00:00</r:ArrivalDate>
                <r:DepartureDate>2006-12-05T00:00:00</r:DepartureDate>
                <r:ShortRateCode>QUEEN</r:ShortRateCode>
                <r:ShortRoomType>QUEEN</r:ShortRoomType>
                <r:ResortId>HTNG1</r:ResortId>
            </r:ReservationData>
        </FetchReservationResponse>
    </soap:Body>
</soap:Envelope>
```

13.3 Post Payment

The post payment message is used to add a charge to the guest folio. This message is generated by a Posting System and sent to the Folio System. In many instances, the Posting System role is performed by the Activity Reservation System and the Folio System is actually handled by the Accommodation Reservation System. The requesting system send a PostPayment (pmtpostingupdate) request, and receives back a PostPaymentResponse (pmtpostingresults) response message.

Port	ReservationProviderPortType
Binding	ReservationProviderBinding
Operation	PostPayment
Soap Action	http://htng.org/PWSWG/2006/08/SingleGuestItinerary#PostPayment
Input	PostPaymentRequest
Output	PostPaymentResponse
Primary Schema	PmtPosting.xsd
Role(s) Implemented	Folio System

13.3.1 PostPaymentRequest



Pmtpostingupdate
<xs:element name="pmtpostingupdate">

```
    <xs:annotation>
    <xs:documentation>pmtpostingupdate as defined by the HTNG working group 2004</xs:documentation>
    </xs:annotation>
    <xs:complexType>
       <xs:sequence>
          <xs:element name="Number">
             <xs:complexType>
                <xs:simpleContent>
                   <xs:extension base="xs:string">
                      <xs:attribute name="type" use="required">
                         <xs:simpleType>
                            <xs:restriction base="xs:NMTOKEN">
                               <xs:enumeration value="unit" />
                               <xs:enumeration value="resno" />
                            </xs:restriction>
                         </xs:simpleType>
                      </xs:attribute>
                   </xs:extension>
                </xs:simpleContent>
             </xs:complexType>
          </xs:element>
          <xs:element name="ClientId" type="xs:string" />
          <xs:element name="PostingGUID" type="xs:string" />
          <xs:element name="ZoomInKey" type="xs:string" />
          <xs:element minOccurs="0" name="AccountDetail" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
             type="q1:AccountDetailType" />
          <xs:element minOccurs="0" name="CreditCardDetail" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
             type="q2:CreditCardDetailType" />
          <xs:element name="VoidFlag" type="xs:boolean" />
          <xs:element name="PostPropertyId" type="xs:unsignedInt" />
          <xs:element name="RegisterId" type="xs:unsignedInt" />
          <xs:element name="CashierEmpId" type="xs:unsignedLong" />
          <xs:element name="ServerEmpId" type="xs:unsignedLong" />
          <xs:element name="RevenueCenterId" type="xs:unsignedLong" />
          <xs:element name="CheckNumber" type="xs:unsignedInt" />
          <xs:element name="Covers" type="xs:int" />
          <xs:element name="MealPeriod" type="xs:unsignedInt" />
          <xs:element name="PaymentId" type="xs:unsignedLong" />
          <xs:element name="TipAmount" type="xs:int" />
          <xs:element name="PaymentAmount" type="xs:int" />
          <xs:element name="CheckAmount" type="xs:int" />
          <xs:element name="ResortId" type="xs:string" />
          <xs:element name="Itemizers" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
             type="q3:ItemizersType" />
          <xs:element minOccurs="0" name="Extensions" xmlns:q4="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
```

```
            type="q4:ExtensionsType" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
Number	element		required	Identifies the account, room, unit, folio, or reservation to post the charge to. The context of the number is determined by the "Type" attribute which is one of *unit* or *resno*.
ClientId	element	string	required	The ID from the requesting system.
PostingGUID	element	string	required	A unique identifier for the posting generated by the requesting system.
ZoomInKey	element	string	required	A key which can be used by the receiving system to acquire additional details for the posting.
AccountDetail	element	AccountDetailType	optional	A record which contains detailed account data such as a particular revenue bucket.
CreditCardDetail	element	CreditCardDetailType	optional	Credit card details.
VoidFlag	element	boolean	required	Flag to indicate refund (reversal) posting.
PostPropertyId	element	unsignedInt	required	Property code.
RegisterId	element	unsignedInt	required	Register ID.
CashierEmpId	element	unsignedLong	required	Cashier employee ID.
ServerEmpId	element	unsignedLong	required	Server employee ID.
RevenueCenterId	element	unsignedLong	required	Revenue center in the POS.
CheckNumber	element	unsignedInt	required	Check number.
Covers	element	int	required	The total number of guests on the check.
MealPeriod	element	unsignedInt	required	Meal period.
PaymentId	element	unsignedLong	required	Payment method ID.
TipAmount	element	int	required	Tip amount.
PaymentAmount	element	int	required	Total payment including tip.
CheckAmount	element	int	required	Check amount.
ResortId	element	string	required	Property ID.
Itemizers	element	ItemizersType	required	A list of financial classes, types, and amounts that represents the breakdown of the financial data for the transaction.
Extensions	element	ExtensionsType	optional	Optional system implemented data.

AccountDetailType

AccountDetailType

```
<xs:complexType name="AccountDetailType">
    <xs:annotation>
    <xs:documentation> Contains the details about the account to be used for posting the payment</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Account" type="xs:string" />
        <xs:element name="Unit" type="xs:string" />
```

```
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Account	element	string	required	Posting account number.
Unit	element	string	required	Room number.

CreditCardDetailType

CreditCardDetailType

```
<xs:complexType name="CreditCardDetailType">
    <xs:annotation>
    <xs:documentation>Holds credit card information</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Account" type="xs:string" />
        <xs:element name="ExpireDate" type="xs:string" />
        <xs:element name="CVV2" type="xs:string" />
        <xs:element name="AuthCode" type="xs:string" />
        <xs:element name="CardSwipeTrack1" type="xs:string" />
        <xs:element name="CardSwipeTrack2" type="xs:string" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Account	element	string	required	Credit card account number.
ExpireDate	element	string	required	Expiration date in YYMM format.
CVV2	element	string	required	CVV2 data.
AuthCode	element	string	required	Authorization code from authorizing agency.
CardSwipeTrack1	element	string	required	MSR track 1 data.
CardSwipeTrack2	element	string	required	MSR track 2 data.

ItemizersType

ItemizersType

```
<xs:complexType name="ItemizersType">
    <xs:sequence>
        <xs:element maxOccurs="unbounded" name="Itemizer" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
            type="q1:ItemizerType" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Itemizer	element	ItemizerType	required / multiple	A collection of itemizer records.

ItemizerType

ItemizerType

```
<xs:complexType name="ItemizerType">
    <xs:simpleContent>
        <xs:extension base="xs:int">
            <xs:attribute name="class" use="required">
                <xs:simpleType>
                    <xs:restriction base="xs:NMTOKEN">
                        <xs:enumeration value="discount" />
                        <xs:enumeration value="gratuity" />
                        <xs:enumeration value="sales" />
                        <xs:enumeration value="serviceCharge" />
                        <xs:enumeration value="tax" />
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="id" type="xs:unsignedInt" use="required" />
            <xs:attribute name="name" type="xs:string" use="required" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
ItemizerType	extension	int		The itemizer type.
class	attribute		required	The financial class. Valid values are *discount*, *gratuity*, *sales*, *serviceCharge*, and *tax*.
id	attribute	unsignedInt	required	The subtype within the class.
name	attribute	string	required	A description of the itemizer value.

ExtensionsType

ExtensionsType

```
<xs:complexType name="ExtensionsType">
    <xs:sequence>
        <xs:element maxOccurs="unbounded" name="Extension" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
            type="q1:ExtensionType" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Extension	element	ExtensionType	required / multiple	

ExtensionType

ExtensionType

```
<xs:complexType name="ExtensionType">
    <xs:choice>
        <xs:element name="MenuItemList" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
            type="q1:MenuItemListType" />
        <xs:element name="ReceiptImage" type="xs:string" />
        <xs:element name="SignatureImage" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
            type="q2:SignatureImageType" />
    </xs:choice>
    <xs:attribute name="class" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="menuitemdetail" />
                <xs:enumeration value="receiptimage" />
                <xs:enumeration value="signatureimage" />
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
class	attribute		required	Identifies the type of extension. One of *menuitemdetail*, *receiptimage*, or *signatureimage*.
MenuItemList	element	MenuItemListType	required	A collection of menuitems.
ReceiptImage	element	string	required	A text image of the printed receipt.
SignatureImage	element	SignatureImageType	required	Signature capture image.

MenuItemListType

MenuItemListType

```
<xs:complexType name="MenuItemListType">
    <xs:sequence>
        <xs:element maxOccurs="unbounded" name="MenuItem" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
            type="q1:MenuItemType" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
MenuItem	element	MenuItemType	required / multiple	A collection of menuitems.

MenuItemType

MenuItemType

```
<xs:complexType name="MenuItemType">
    <xs:sequence>
```

```
        <xs:element name="id" type="xs:string" />
        <xs:element name="name" type="xs:string" />
        <xs:element name="qty" type="xs:int" />
        <xs:element name="price" type="xs:int" />
        <xs:element name="extPrice" type="xs:int" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
id	element	string	required	The menu item identifier.
name	element	string	required	The name of the menu item.
qty	element	int	required	The quanty sold.
price	element	int	required	The menu price for a single item.
extPrice	element	int	required	Extension of quantity * price.

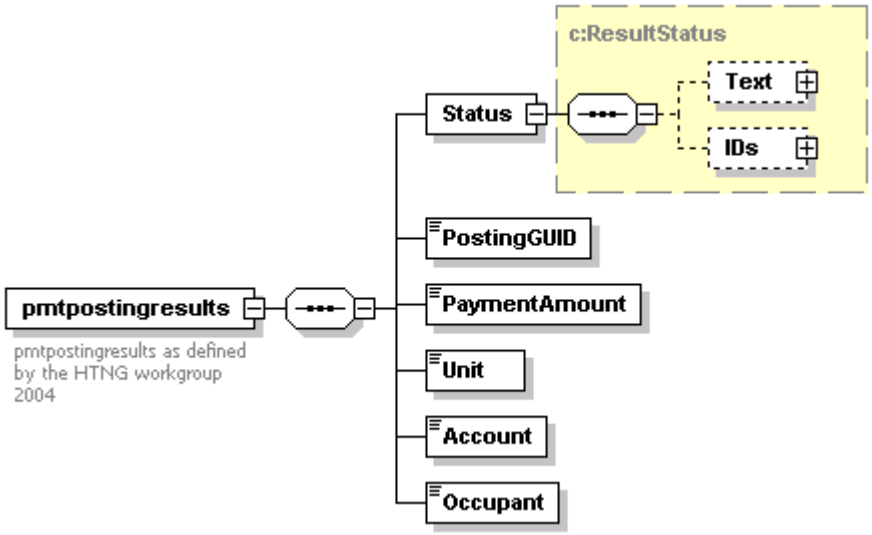
SignatureImageType

SignatureImageType

```
<xs:complexType name="SignatureImageType">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="encoding" type="xs:string" use="required" />
            <xs:attribute name="imageFormat" type="xs:string" use="required" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
SignatureImageType	extension	string		Contains the signature data.
encoding	attribute	string	required	Defines the format of the data.
imageFormat	attribute	string	required	Defines the format of the data.

13.3.2 PostPaymentResponse



Pmtpostingresults

```
<xs:element name="pmtpostingresults">
    <xs:annotation>
    <xs:documentation>pmtpostingresults as defined by the HTNG workgroup 2004</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Status" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:ResultStatus" />
            <xs:element name="PostingGUID" type="xs:string" />
            <xs:element name="PaymentAmount" type="xs:int" />
            <xs:element name="Unit" type="xs:string" />
            <xs:element name="Account" type="xs:string" />
            <xs:element name="Occupant" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
Status	element	ResultStatus	required	The result of the posting.
PostingGUID	element	string	required	Return of the submitted GUID
PaymentAmount	element	int	required	The total payment amount.
Unit	element	string	required	The unit posted to.
Account	element	string	required	The account posted to.

Occupant	element	string	required	The name of the occupant.

13.3.3 Sample Message

[*Request*]


```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#PostPayment</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:SPASOFT</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:a57205c6-61b4-47ab-aef7-424ab4c0ac5e</wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.micros.com/HTNGActivity/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-4b9231cb-8bfd-468a-8380-9b38e4191ad2">
                <wsu:Created>2006-11-02T19:52:58Z</wsu:Created>
                <wsu:Expires>2006-11-02T19:57:58Z</wsu:Expires>
            </wsu:Timestamp>
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
d4f5d827-3e6f-47f9-a510-540377f74faa">
                <wsse:Username>HTNG</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">HTNG</wsse:Password>
                <wsse:Nonce>eceBCgHaKyvv2r8c6JaMwg==</wsse:Nonce>
                <wsu:Created>2006-11-02T19:52:58Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <pmtpostingupdate xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">
            <Number type="resno">1620979</Number>
            <ClientId>604</ClientId>
            <PostingGUID>821072F0360F44B1A583F94F16D252E</PostingGUID>
            <ZoomInKey>1354</ZoomInKey>
            <AccountDetail>
                <Account>1620979</Account>
                <Unit>10</Unit>
            </AccountDetail>
```

```
            <VoidFlag>false</VoidFlag>
            <PostPropertyId>0</PostPropertyId>
            <RegisterId>8</RegisterId>
            <CashierEmpId>98</CashierEmpId>
            <ServerEmpId>98</ServerEmpId>
            <RevenueCenterId>1</RevenueCenterId>
            <CheckNumber>1443</CheckNumber>
            <Covers>1</Covers>
            <MealPeriod>1</MealPeriod>
            <PaymentId>5</PaymentId>
            <TipAmount>4091</TipAmount>
            <PaymentAmount>5113</PaymentAmount>
            <CheckAmount>5113</CheckAmount>
            <ResortId>HTNG1</ResortId>
            <Itemizers>
                <Itemizer id="2" name="Spa Activities" class="sales">4091</Itemizer>
                <Itemizer id="1" name="serviceCharge" class="serviceCharge">409</Itemizer>
                <Itemizer id="1" name="tax1" class="tax">286</Itemizer>
                <Itemizer id="2" name="tax2" class="tax">327</Itemizer>
            </Itemizers>
            <Extensions>
                <Extension class="menuitemdetail">
                    <MenuItemList>
                        <MenuItem>
                            <id>49</id>
                            <name>Botanical Bath</name>
                            <qty>100</qty>
                            <price>4500</price>
                            <extPrice>5113</extPrice>
                        </MenuItem>
                    </MenuItemList>
                </Extension>
            </Extensions>
        </pmtpostingupdate>
    </soap:Body>
</soap:Envelope>
```


[*Response*]


```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#PostPaymentResponse</wsa:Action>
```

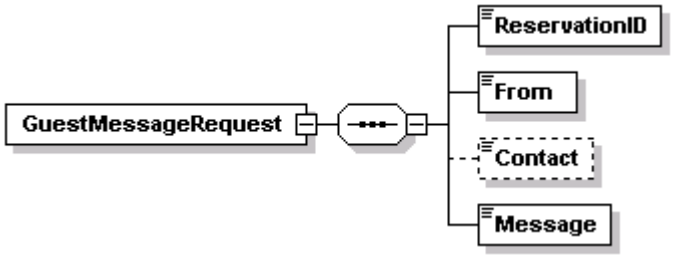
```
        <wsa:MessageID>urn:uuid:2676f529-5f9a-4214-a1da-b9320f615b0c</wsa:MessageID>
        <wsa:RelatesTo>urn:uuid:a57205c6-61b4-47ab-aef7-424ab4c0ac5e</wsa:RelatesTo>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-c52ca723-8c70-45a7-a80e-012543bf3feb">
                <wsu:Created>2006-11-02T19:51:44Z</wsu:Created>
                <wsu:Expires>2006-11-02T19:56:44Z</wsu:Expires>
            </wsu:Timestamp>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <pmtpostingresults xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types">
            <Status resultStatusFlag="SUCCESS"/>
            <PostingGUID>821072F0360F44B1A583F94F16D252E</PostingGUID>
            <PaymentAmount>5113</PaymentAmount>
            <Unit>10</Unit>
            <Account>1620979</Account>
        </pmtpostingresults>
    </soap:Body>
</soap:Envelope>
```


13.4 Guest Message

The guest message function is provided in order to send a message to a guest. This is typically provided by the Accommodation Reservation System and may generate a guest notification, such as turning on the message light in the guest's room.

Port	ReservationProviderPortType
Binding	ReservationProviderBinding
Operation	GuestMessage
Soap Action	http://htng.org/PWSWG/2006/08/SingleGuestItinerary#GuestMessage
Input	GuestMessageRequest
Output	GuestMessageResponse
Primary Schema	Reservation.xsd
Role(s) Implemented	Accommodation Reservation System

13.4.1 GuestMessageRequest

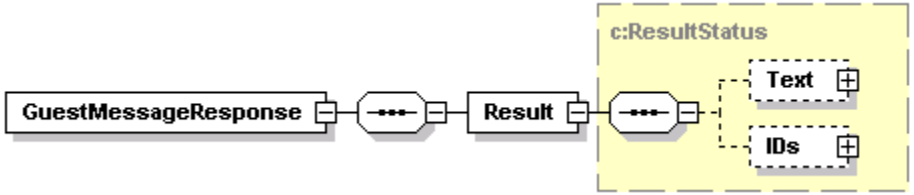


GuestMessageRequest

```
<xs:element name="GuestMessageRequest">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ReservationID" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueID" />
            <xs:element name="From" type="xs:string" />
            <xs:element minOccurs="0" name="Contact" type="xs:string" />
            <xs:element name="Message">
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:extension base="xs:string">
                            <xs:attribute name="notify" type="xs:boolean" use="optional" />
                        </xs:extension>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
ReservationID	element	UniqueID	required	The reservation ID in the receiving system.
From	element	string	required	Identifies who the message is from.
Contact	element	string	optional	Identifies a contact to respond to.
Message	element	complex	required	The message text and includes a flag to indicate whether a notification should be sent to the guest (for example, to turn on the message light in the guest's room).

13.4.2 GuestMessageResponse



GuestMessageResponse				
<xs:element name="GuestMessageResponse">
<xs:complexType>
<xs:sequence>
<xs:element name="Result" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:ResultStatus" />
</xs:sequence>
</xs:complexType>
</xs:element>				
Name	**Type**	**Data Type**	**Use**	**Comments**
Result	element	ResultStatus	required	The result from the operation.

13.4.3 Sample Message

[*Request*]

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#GuestMessage</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:SPASOFT</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:dc7b1331-9a8b-4fac-af05-8769dd4c8647</wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.micros.com/HTNGActivity/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-879a2d3a-aac3-4d47-9143-f63507e7fedb">
                <wsu:Created>2007-01-05T16:19:03Z</wsu:Created>
                <wsu:Expires>2007-01-05T16:24:03Z</wsu:Expires>
            </wsu:Timestamp>
```

```
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
ad5a7c5b-dc6e-4526-9e46-d98d4e7980f3">
                <wsse:Username>HTNG</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">HTNG</wsse:Password>
                <wsse:Nonce>7QlxcSWRwFhg4IjYlWcpHg==</wsse:Nonce>
                <wsu:Created>2007-01-05T16:19:03Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <GuestMessageRequest xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
xmlns:a="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">
            <ReservationID source="OPERA">962887</ReservationID>
            <From>Tranquility Spa, Palm Springs, CA</From>
            <Contact>Tripp, Debbie</Contact>
            <Message notify="true">Dear Ms. Tripp, Thank you for visiting our Spa! We hope to see you again soon!</Message>
        </GuestMessageRequest>
    </soap:Body>
</soap:Envelope>
```


[*Response*]


```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#GuestMessageResponse</wsa:Action>
        <wsa:MessageID>urn:uuid:029efdba-e51e-4c44-82e0-96195e0ca942</wsa:MessageID>
        <wsa:RelatesTo>urn:uuid:dc7b1331-9a8b-4fac-af05-8769dd4c8647</wsa:RelatesTo>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-c00eea4b-e3dc-4b4f-b21c-8754593fa33e">
                <wsu:Created>2007-01-05T16:21:47Z</wsu:Created>
                <wsu:Expires>2007-01-05T16:26:47Z</wsu:Expires>
            </wsu:Timestamp>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <GuestMessageResponse xmlns:r="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
xmlns:n="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
xmlns:a="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
```

```
xmlns:p="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types">
            <r:Result resultStatusFlag="SUCCESS"/>
        </GuestMessageResponse>
    </soap:Body>
</soap:Envelope>
```

Chapter 14 Reservation Sync Port Type

The reservation sync port is provided to allow Accommodation Reservation System to notify of a change in the status of a guest. For instance, the Activity Reservation System may desire to know when a guest has checked in or checked out. There is only one function defined:

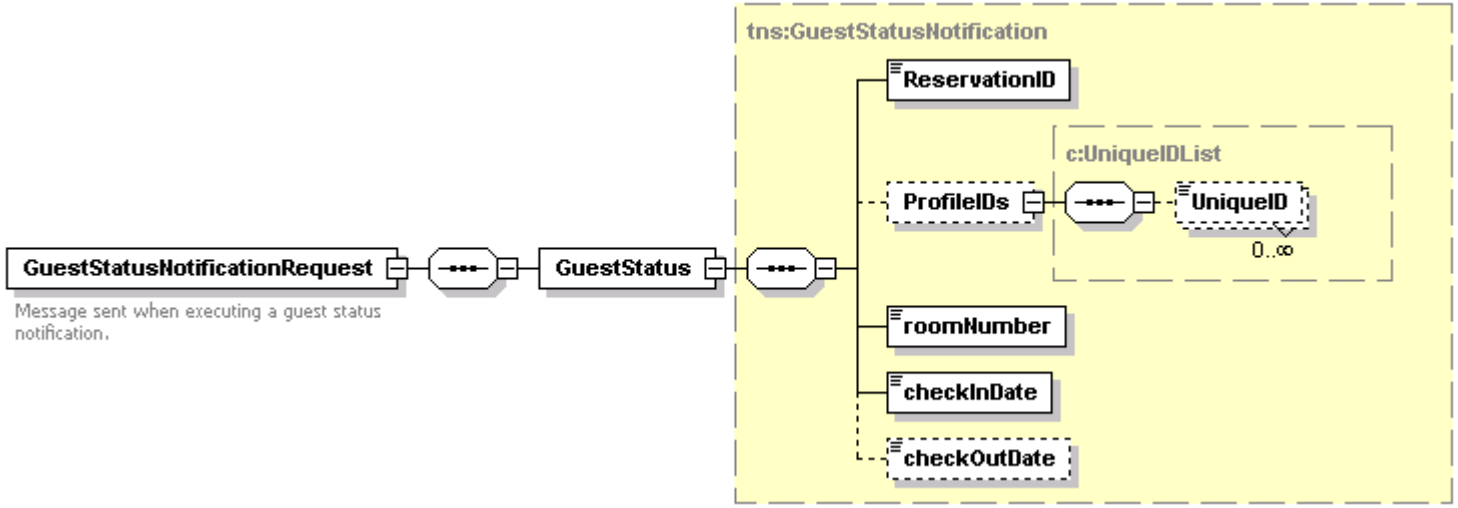
- GuestStatusNotification
  - This message may be send by the Accommodation Reservation System to notify various changes in a guest's status.

14.1 Guest Status Notification

The guest status notification message is generated by the Accommodation Reservation System to notify the Activity Reservation System of a change in the status of a guest. Status changes such as Checked In or Checked Out may be sent.

Port	ReservationSyncPortType
Binding	ReservationSyncBinding
Operation	GuestStatusNotification
Soap Action	http://htng.org/PWSWG/2006/08/SingleGuestItinerary#GuestStatusNotification
Input	GuestStatusNotificationRequest
Output	GuestStatusNotificationResponse
Primary Schema	Reservation.xsd
Role(s) Implemented	Activity Reservation System

14.1.1 GuestStatusNotificationRequest



GuestStatusNotificationRequest

```
<xs:element name="GuestStatusNotificationRequest">
    <xs:annotation>
    <xs:documentation>Message sent when executing a guest status notification.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="GuestStatus" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
                type="q1:GuestStatusNotification" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
GuestStatus	element	GuestStatusNotification	required	The guest status record.

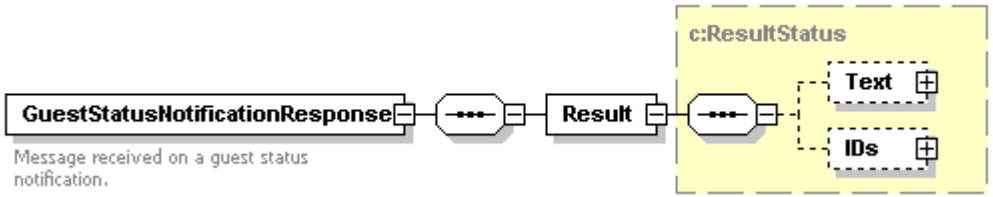
GuestStatusNotification

GuestStatusNotification

```
<xs:complexType name="GuestStatusNotification">
    <xs:sequence>
        <xs:element name="ReservationID" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueID" />
        <xs:element minOccurs="0" name="ProfileIDs" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q2:UniqueIDList" />
        <xs:element name="roomNumber" type="xs:string" />
        <xs:element name="checkInDate" type="xs:dateTime" />
        <xs:element minOccurs="0" name="checkOutDate" type="xs:dateTime" />
    </xs:sequence>
    <xs:attribute name="reservationStatus" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
        type="q3:ReservationStatusType" use="required" />
    <xs:attribute name="otherReservationStatus" type="xs:string" use="optional" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
reservationStatus	attribute	ReservationStatusType	required	The reservation status
otherReservationStatus	attribute	string	optional	Not used.
ReservationID	element	UniqueID	required	The reservation ID as defined by the requesting system.
ProfileIDs	element	UniqueIDList	optional	The primary guest profile ID.
roomNumber	element	string	required	The guest's room number.
checkInDate	element	dateTime	required	The arrival date.
checkOutDate	element	dateTime	optional	The departure date.

14.1.2 GuestStatusNotificationResponse



GuestStatusNotificationResponse

```
<xs:element name="GuestStatusNotificationResponse">
    <xs:annotation>
    <xs:documentation>Message received on a guest status notification.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Result" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:ResultStatus" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
Result	element	ResultStatus	required	The result from the operation.

14.1.3 Sample Message

[*Request*]

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#GuestStatusNotification</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:OPERA</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:39535817-b579-427d-9e19-fb97c2ee51cb</wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.springermiller.com/HTNG_2_0/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-959e874e-9adb-438a-aa6d-27d7616b11a1">
```

```
            <wsu:Created>2007-02-06T15:31:11Z</wsu:Created>
            <wsu:Expires>2007-02-06T15:36:11Z</wsu:Expires>
        </wsu:Timestamp>
        <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
3e469327-ccf2-4dbf-a7c2-84fdcbc1d2ea">
            <wsse:Username>OPERA</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">OPERA</wsse:Password>
            <wsse:Nonce>vbBf4EUhxiEJkxIPkZHedg==</wsse:Nonce>
            <wsu:Created>2007-02-06T15:31:11Z</wsu:Created>
        </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <GuestStatusNotificationRequest xmlns:r="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
xmlns:n="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
xmlns:a="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
xmlns:p="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/PaymentPosting/Types"
xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types">
        <r:GuestStatus reservationStatus="RESERVED">
            <r:ReservationID source="OPERA">1003385</r:ReservationID>
            <r:ProfileIDs>
                <c:UniqueID source="OPERA">5617209</c:UniqueID>
            </r:ProfileIDs>
            <r:checkInDate>2007-02-04T00:00:00</r:checkInDate>
            <r:checkOutDate>2007-02-07T00:00:00</r:checkOutDate>
        </r:GuestStatus>
    </GuestStatusNotificationRequest>
  </soap:Body>
</soap:Envelope>
```


[*Response*]


```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
  <soap:Header>
    <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#GuestStatusNotificationResponse</wsa:Action>
    <wsa:MessageID>urn:uuid:54a0d7a6-5a44-4075-aa79-8b3de55b4ae4</wsa:MessageID>
    <wsa:RelatesTo>urn:uuid:39535817-b579-427d-9e19-fb97c2ee51cb</wsa:RelatesTo>
    <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
    <wsse:Security>
        <wsu:Timestamp wsu:Id="Timestamp-883fbd19-c3cf-48fa-8649-302d550cddaa">
```

```
            <wsu:Created>2007-02-06T15:32:43Z</wsu:Created>
            <wsu:Expires>2007-02-06T15:37:43Z</wsu:Expires>
        </wsu:Timestamp>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <GuestStatusNotificationResponse xmlns="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Reservation/Types"
xmlns:a="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
xmlns:c="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types">
        <Result code="" resultStatusFlag="SUCCESS">
            <c:Text>
                <c:TextElement language="en"/>
            </c:Text>
            <c:IDs>
                <c:UniqueID source="OPERA">5617209</c:UniqueID>
            </c:IDs>
        </Result>
    </GuestStatusNotificationResponse>
  </soap:Body>
</soap:Envelope>
```


14.2 Fetch Check Details

This message, FetchCheckDetails, was inadvertently included in the first release of this document, and was not intended to be a part of the 2008A release. It is expected to be a part of the 2008B release. This note has been added, rather than removing the message from this document, in case any implementations are already using it.

The fetch check details message is generated by the Accommodation Reservation System to provide the details of a guest check image. The requesting system submits a FetchCheckDetailsRequest message, and the responding system returns a FetchChecketailsResponse message. If successful, the response message will include the details of the guest check image.

Port	ReservationSyncPortType
Binding	ReservationSyncBinding
Operation	FetchCheckDetails
Soap Action	http://htng.org/PWS/2008A/SingleGuestItinerary#FetchCheckDetails
Input	FetchCheckDetailsRequest
Output	FetchCheckDetailsResponse
Primary Schema	Reservation.xsd
Role(s) Implemented	Accommodation Reservation System

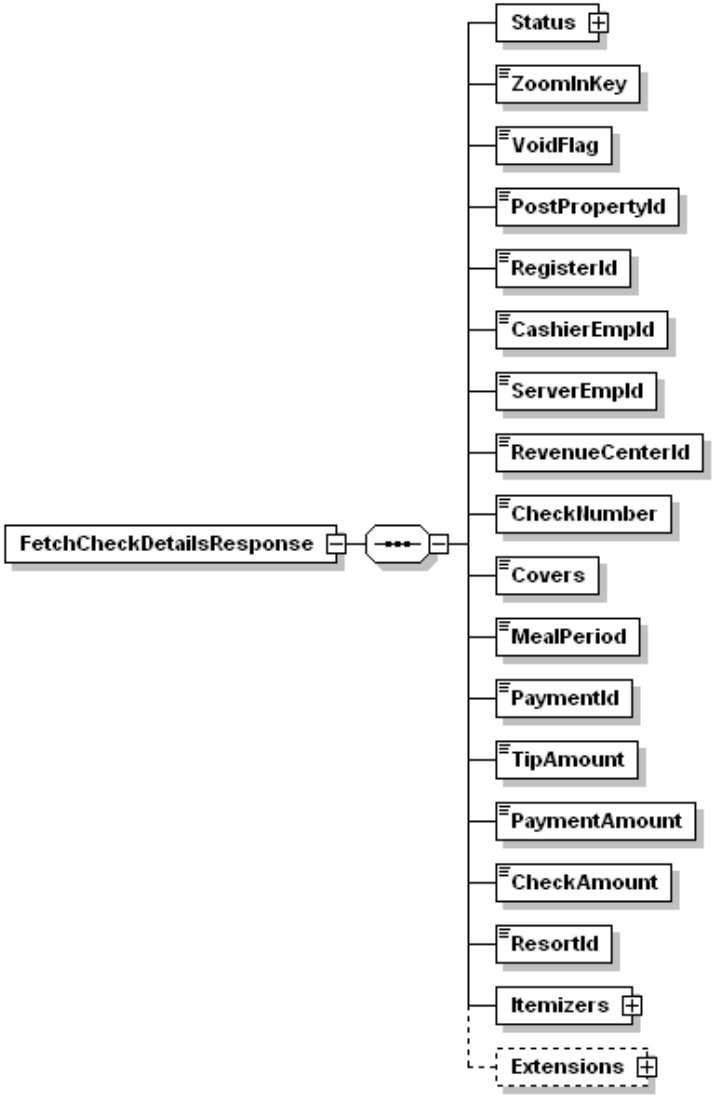
14.2.1 FetchCheckDetailsRequest

FetchCheckDetailsRequest

```
<xs:element name="FetchCheckDetailsRequest">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ZoomInKey" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
ZoomInKey	element	string	required	A key which can be used by the receiving system to acquire additional details for the check.

14.2.2 FetchCheckDetailsResponse



FetchCheckDetailsResponse

```
<xs:element name="FetchCheckDetailsResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Status" xmlns:q1="http://htng.org/PWS/2008A/SingleGuestItinerary/Common/Types" type="q1:ResultStatus" />
            <xs:element name="ZoomInKey" type="xs:string" />
            <xs:element name="VoidFlag" type="xs:boolean" />
            <xs:element name="PostPropertyId" type="xs:unsignedInt" />
            <xs:element name="RegisterId" type="xs:unsignedInt" />
            <xs:element name="CashierEmpId" type="xs:unsignedLong" />
            <xs:element name="ServerEmpId" type="xs:unsignedLong" />
            <xs:element name="RevenueCenterId" type="xs:unsignedLong" />
            <xs:element name="CheckNumber" type="xs:unsignedInt" />
            <xs:element name="Covers" type="xs:int" />
            <xs:element name="MealPeriod" type="xs:unsignedInt" />
            <xs:element name="PaymentId" type="xs:unsignedLong" />
            <xs:element name="TipAmount" type="xs:int" />
            <xs:element name="PaymentAmount" type="xs:int" />
            <xs:element name="CheckAmount" type="xs:int" />
            <xs:element name="ResortId" type="xs:string" />
            <xs:element name="Itemizers" xmlns:q2="http://htng.org/PWS/2008A/SingleGuestItinerary/PaymentPosting/Types" type="q2:ItemizersType" />
            <xs:element minOccurs="0" name="Extensions" xmlns:q3="http://htng.org/PWS/2008A/SingleGuestItinerary/PaymentPosting/Types"
                type="q3:ExtensionsType" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
Status	element	ResultStatus	required	An indication of the success or failure of the called function.
ZoomInKey	element	string	required	A key which can be used by the receiving system to acquire additional details for the check.
VoidFlag	element	boolean	required	Flag to indicate refund (reversal) posting.
PostPropertyId	element	unsignedInt	required	The ID of the posting property.
RegisterId	element	unsignedInt	required	The register ID code.
CashierEmpId	element	unsignedLong	required	The the cashier ID code.
ServerEmpId	element	unsignedLong	required	The server employee ID code.
RevenueCenterId	element	unsignedLong	required	The revenue center ID code in the POS.
CheckNumber	element	unsignedInt	required	The check number.
Covers	element	int	required	The number of people on the check.
MealPeriod	element	unsignedInt	required	The meal period, such as breakfast, lunch, or dinner.
PaymentId	element	unsignedLong	required	The payment ID code.
TipAmount	element	int	required	The tip amount.
PaymentAmount	element	int	required	The payment amount.
CheckAmount	element	int	required	The check amount.
ResortId	element	string	required	The resort ID code.

Itemizers	element	ItemizersType	required	The number of items on the check.
Extensions	element	ExtensionsType	optional	Optional system implemented data.

14.2.3 Sample Messages

[*Request*]

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWS/2008A//SingleGuestItinerary#FetchCheckDetailsRequest</wsa:Action>
        <wsa:From>
            <wsa:Address>urn:SPASOFT</wsa:Address>
        </wsa:From>
        <wsa:MessageID>urn:uuid:261335f2-bd1d-49e6-96b4-8fcdf2137f8a</wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://htng.org/PWS/2008A/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>http://www.micros.com/HTNGActivity/</wsa:To>
        <wsse:Security soap:mustUnderstand="1">
            <wsu:Timestamp wsu:Id="Timestamp-d99e21bf-1227-410c-bea3-d8705adcf032">
                <wsu:Created>2006-11-02T20:22:14Z</wsu:Created>
                <wsu:Expires>2006-11-02T20:27:14Z</wsu:Expires>
            </wsu:Timestamp>
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="SecurityToken-6f2c0e7f-ac98-4ea3-a726-d67076fc9c72">
                <wsse:Username>HTNG</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">HTNG</wsse:Password>
                <wsse:Nonce>IWGqOAykiUQXnWN5x1wS9Q==</wsse:Nonce>
                <wsu:Created>2006-11-02T20:22:14Z</wsu:Created>
            </wsse:UsernameToken>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <FetchCheckDetailsRequest xmlns="http://htng.org/PWS/2008A/SingleGuestItinerary/PaymentPosting/Types"
        xmlns:c="http://htng.org/PWS/2008A/SingleGuestItinerary/Common/Types"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://htng.org/PWS/2008A/SingleGuestItinerary/PaymentPosting/Types">
            <ZoomInKey>1234</ZoomInKey>
        </FetchCheckDetailsRequest>
```

```
    </soap:Body>
</soap:Envelope>
```

[*Response*]


```xml
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <soap:Header>
        <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#FetchCheckDetailsResponse</wsa:Action>
        <wsa:MessageID>urn:uuid:114edffa-4634-49fe-93af-a0431ab0f46b</wsa:MessageID>
        <wsa:RelatesTo>urn:uuid:261335f2-bd1d-49e6-96b4-8fcdf2137f8a</wsa:RelatesTo>
        <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
        <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-4f4401bc-e095-4db8-8f34-eb53bd8a656f">
                <wsu:Created>2006-11-02T20:20:52Z</wsu:Created>
                <wsu:Expires>2006-11-02T20:25:52Z</wsu:Expires>
            </wsu:Timestamp>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <FetchCheckDetailsResponse xmlns="http://htng.org/PWS/2008A/SingleGuestItinerary/PaymentPosting/Types"
xmlns:c="http://htng.org/PWS/2008A/SingleGuestItinerary/Common/Types" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://htng.org/PWS/2008A/SingleGuestItinerary/PaymentPosting/Types">
            <Status resultStatusFlag="SUCCESS">
                <c:Text>
                    <c:TextElement language="en-us">en</c:TextElement>
                </c:Text>
                <c:IDs>
                    <c:UniqueID source="String">1234</c:UniqueID>
                </c:IDs>
            </Status>
            <ZoomInKey>1234</ZoomInKey>
            <VoidFlag>1</VoidFlag>
            <PostPropertyId>4</PostPropertyId>
            <RegisterId>23</RegisterId>
            <CashierEmpId>4294967295</CashierEmpId>
            <ServerEmpId>4294967295</ServerEmpId>
            <RevenueCenterId>4294967295</RevenueCenterId>
            <CheckNumber>12560</CheckNumber>
            <Covers>2</Covers>
            <MealPeriod>1</MealPeriod>
```

```
        <PaymentId>4294967295</PaymentId>
        <TipAmount>1</TipAmount>
        <PaymentAmount>12</PaymentAmount>
        <CheckAmount>12</CheckAmount>
        <ResortId>CASINO</ResortId>
        <Itemizers>
            <Itemizer class="discount" id="1" name="FOOD">2</Itemizer>
        </Itemizers>
        <Extensions>
            <Extension class="menuitemdetail">
                <MenuItemList>
                    <MenuItem>
                        <id>10</id>
                        <name>Appetizer</name>
                        <qty>1</qty>
                        <price>10</price>
                        <extPrice>12</extPrice>
                    </MenuItem>
                </MenuItemList>
            </Extension>
        </Extensions>
    </FetchCheckDetailsResponse>
  </soap:Body>
</soap:Envelope>
```

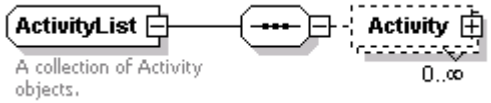
Chapter 15 Common Data Elements

The Activity Web Service messages share several common data elements. These are consolidated here for reference.

15.1 Activity Schema Elements (Activity.xsd)

Namespace	http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types

15.1.1 ActivityList

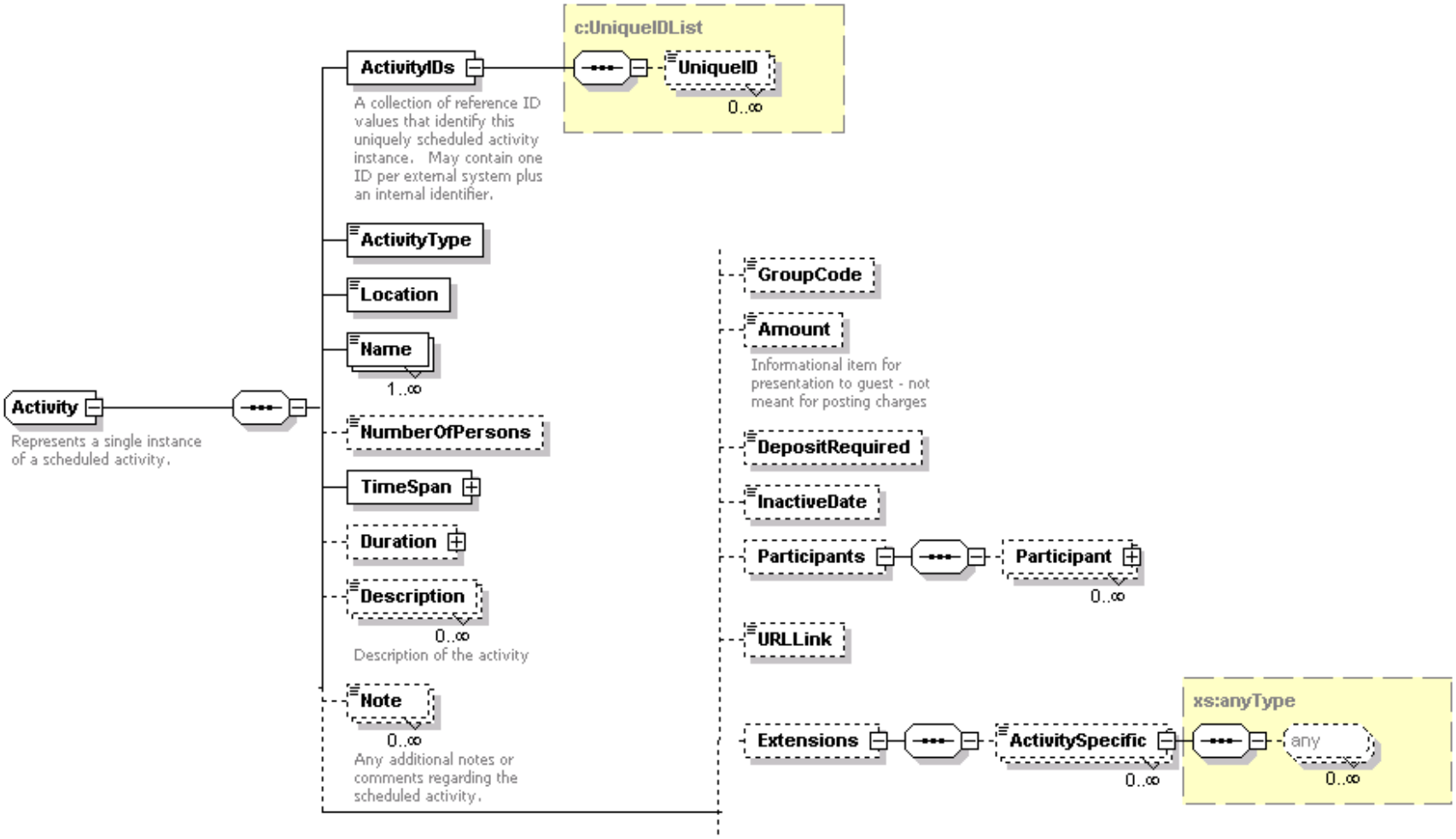


ActivityList

```
<xs:complexType name="ActivityList">
    <xs:annotation>
    <xs:documentation xml:lang="en">A collection of Activity objects.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Activity">
            <xs:complexType>
                <xs:complexContent mixed="false">
                    <xs:extension xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types" base="q1:Activity" />
                </xs:complexContent>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Activity	element		optional / multiple	A collection (zero or more) of Activity elements.

15.1.2 Activity



Activity
<xs:complexType name="Activity"> <xs:annotation> <xs:documentation xml:lang="en">Represents a single instance of a scheduled activity.</xs:documentation>

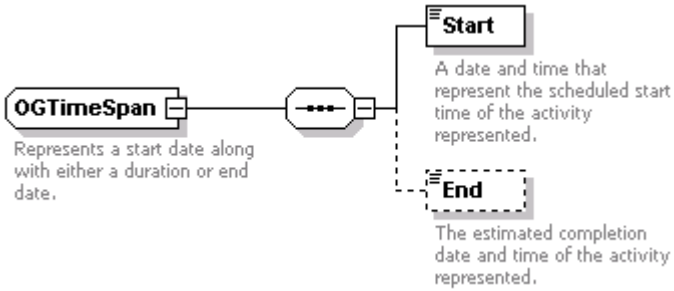
```
    </xs:annotation>
    <xs:sequence>
        <xs:element name="ActivityIDs" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueIDList">
            <xs:annotation>
            <xs:documentation xml:lang="en">A collection of reference ID values that identify this uniquely scheduled activity instance.   May contain one ID
                per external system plus an internal identifier.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="ActivityType" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types" type="q2:ActivityType" />
        <xs:element name="Location" type="xs:string" />
        <xs:element maxOccurs="unbounded" name="Name" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q3:Text" />
        <xs:element minOccurs="0" name="NumberOfPersons" type="xs:int" />
        <xs:element name="TimeSpan" xmlns:q4="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types" type="q4:OGTimeSpan" />
        <xs:element minOccurs="0" name="Duration" xmlns:q5="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Activity/Types"
            type="q5:OGTimeSpan" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Description"
            xmlns:q6="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q6:Text">
            <xs:annotation>
            <xs:documentation>Description of the activity </xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Note"
            xmlns:q7="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q7:Text">
            <xs:annotation>
            <xs:documentation xml:lang="en">Any additional notes or comments regarding the scheduled activity.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="GroupCode" type="xs:string" />
        <xs:element minOccurs="0" name="Amount" xmlns:q8="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q8:Amount">
            <xs:annotation>
            <xs:documentation>Informational item for presentation to guest - not meant for posting charges</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="DepositRequired">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension xmlns:q9="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" base="q9:Amount">
                        <xs:attribute name="collectedBy">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:enumeration value="Vendor" />
                                    <xs:enumeration value="Agent" />
                                    <xs:enumeration value="Other" />
                                </xs:restriction>
```

```
                    </xs:simpleType>
                </xs:attribute>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element minOccurs="0" name="InactiveDate" type="xs:dateTime" />
<xs:element minOccurs="0" name="Participants">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="Participant"
                xmlns:q10="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q10:PersonName" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element minOccurs="0" name="URLLink" type="xs:string" />
<xs:element minOccurs="0" name="Extensions">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="ActivitySpecific" type="xs:anyType" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="status" type="xs:string" use="required" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
status	attribute	string	required	Activity reservation status. Each system may define its own set of values, but typical examples are RESERVED, BOOKED, CANCELLED, and TENTATIVE.
ActivityIDs	element	UniqueIDList	required	One or more ID's that identify the activity record. The source value indicates the creator/owner of the ID.
ActivityType	element	ActivityType	required	The type of activity. Defined values in the schema include: *Golf*, *Class*, *Spa*, *TableReservation*, *Tour*, and *Tennis*. The schema may be extended to include additional values.
Location	element	string	required	The location for the activity. This is implemented as a location code rather than descriptive text.
Name	element	Text	required / multiple	A short descriptive name for the activity.
NumberOfPersons	element	int	optional	The number of participants scheduled for the activity.
TimeSpan	element	OGTimeSpan	required	The scheduled start time and length of the activity.
Duration	element	OGTimeSpan	optional	The length of an activity
Description	element	Text	optional / multiple	The long description for the activity.
Note	element	Text	optional /	Additional notes about the activity.

			multiple	
GroupCode	element	string	optional	If applicable, the group code for the activity (for example, in an accommodation reservation system, the allotment or block code).
Amount	element	Amount	optional	The amount charged for the activity.
DepositRequired	element	Amount (extension)	optional	The deposit amount collected or required for the activity. This element also includes an identification of the responsible party to collect the deposit as either: *Vendor*, *Agent* or *Other*.
InactiveDate	element	dateTime	optional	The timestamp applicable in case the activity is cancelled or deleted. If set, the activity should be considered removed.
Participants	element	PersonName	optional / multiple	The names of the participants.
URLLink	element	string	optional	The URL link to launch the Activity Reservation System to view or edit full details of this activity.
Extensions	element		optional	Vendor extensions to the activity record.

15.1.3 OGTimeSpan



OGTimeSpan

```
<xs:complexType name="OGTimeSpan">
    <xs:annotation>
    <xs:documentation xml:lang="en">Represents a start date along with either a duration or end date.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="Start" type="xs:dateTime">
            <xs:annotation>
            <xs:documentation xml:lang="en">A date and time that represent the scheduled start time of the activity represented.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element minOccurs="0" name="End" type="xs:dateTime">
            <xs:annotation>
            <xs:documentation xml:lang="en">The estimated completion date and time of the activity represented.</xs:documentation>
            </xs:annotation>
        </xs:element>
```

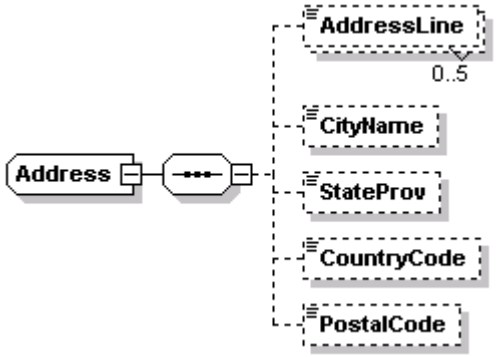
```
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Start	element	dateTime	required	The start date and time.
End	element	dateTime	optional	The optional completion date and time.

15.2 Common Schema Elements (Common.xsd)

Namespace	http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types

15.2.1 Address



Address

```
<xs:complexType name="Address">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="5" name="AddressLine" type="xs:string" />
        <xs:element minOccurs="0" name="CityName" type="xs:string" />
        <xs:element minOccurs="0" name="StateProv" type="xs:string" />
        <xs:element minOccurs="0" name="CountryCode" type="xs:string" />
        <xs:element minOccurs="0" name="PostalCode" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="addressType" type="xs:string" />
    <xs:attribute name="otherAddressType" type="xs:string" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
addressType	attribute	string	none	Type of address. Typical values are HOME or BUSINESS, but any internally defined value may be used.
otherAddressType	attribute	string	none	Not currently used.
AddressLine	element	string	optional /	Up to five address lines may be specified.

			multiple	
CityName	element	string	optional	City.
StateProv	element	string	optional	State.
CountryCode	element	string	optional	Country.
PostalCode	element	string	optional	Postal code.

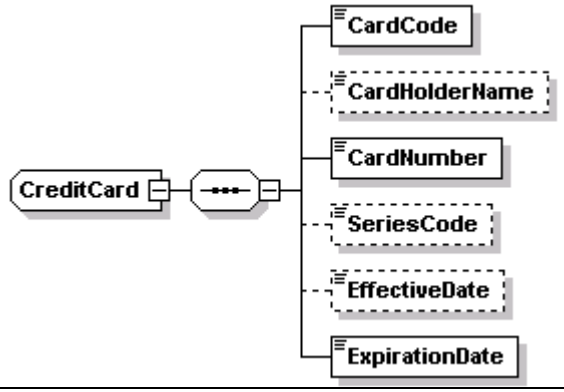
15.2.2 Amount

Amount

```
<xs:complexType name="Amount">
    <xs:simpleContent>
        <xs:extension base="xs:double">
            <xs:attribute name="currencyCode" type="xs:string" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Amount	extension	double		The denominational amount.
currencyCode	attribute	string	none	The currency code the amount is expressed in.

15.2.3 Credit Card



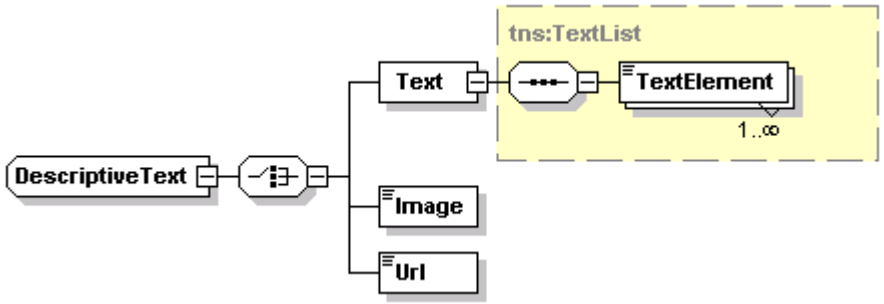
CreditCard

```
<xs:complexType name="CreditCard">
    <xs:sequence>
        <xs:element name="CardCode" type="xs:string" />
        <xs:element minOccurs="0" name="CardHolderName" type="xs:string" />
        <xs:element name="CardNumber" type="xs:string" />
        <xs:element minOccurs="0" name="SeriesCode" type="xs:string" />
```

```
        <xs:element minOccurs="0" name="EffectiveDate" type="xs:date" />
        <xs:element name="ExpirationDate" type="xs:date" />
    </xs:sequence>
    <xs:attribute name="cardType" type="xs:string" />
    <xs:attribute name="otherCardType" type="xs:string" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
cardType	attribute	string	none	Credit card type.
otherCardType	attribute	string	none	Not used.
CardCode	element	string	required	Credit card code.
CardHolderName	element	string	optional	Card holder name.
CardNumber	element	string	required	Credit card number.
SeriesCode	element	string	optional	Not used.
EffectiveDate	element	date	optional	Effective date.
ExpirationDate	element	date	required	Expiration date.

15.2.4 Descriptive Text

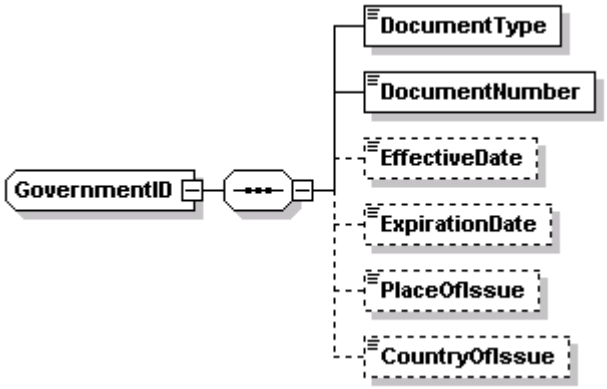


DescriptiveText

```
<xs:complexType name="DescriptiveText">
    <xs:choice>
        <xs:element name="Text" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:TextList" />
        <xs:element name="Image" type="xs:string" />
        <xs:element name="Url" type="xs:anyURI" />
    </xs:choice>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Text	element	TextList	required	Collection of Text elements (Choice)
Image	element	string	required	Image value (Choice).
Url	element	anyURI	required	URL (Choice).

15.2.5 GovernmentID

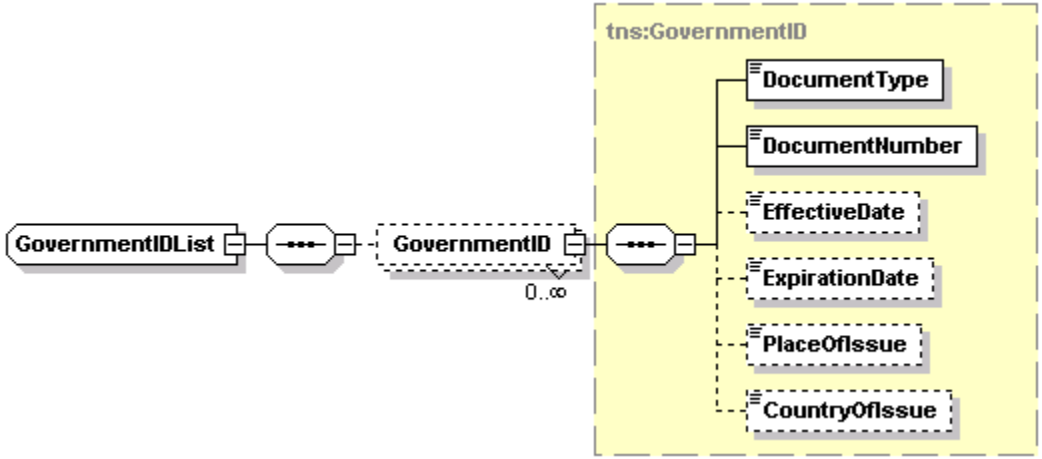


GovernmentID

```
<xs:complexType name="GovernmentID">
    <xs:sequence>
        <xs:element name="DocumentType" type="xs:string" />
        <xs:element name="DocumentNumber" type="xs:string" />
        <xs:element minOccurs="0" name="EffectiveDate" type="xs:date" />
        <xs:element minOccurs="0" name="ExpirationDate" type="xs:date" />
        <xs:element minOccurs="0" name="PlaceOfIssue" type="xs:string" />
        <xs:element minOccurs="0" name="CountryOfIssue" type="xs:string" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
DocumentType	element	string	required	The type of document (e.g. PASSPORT).
DocumentNumber	element	string	required	The document number.
EffectiveDate	element	date	optional	The effective date of the document.
ExpirationDate	element	date	optional	The expiration date of the document.
PlaceOfIssue	element	string	optional	The place of issue.
CountryOfIssue	element	string	optional	The country of issue.

15.2.6 GovernmentIDList

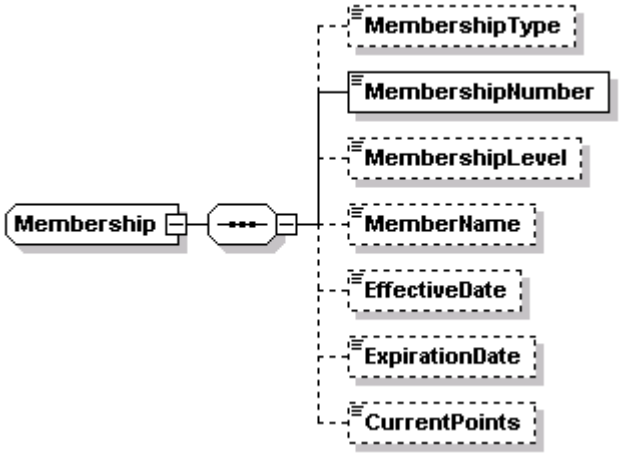


GovernmentIDList

```
<xs:complexType name="GovernmentIDList">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="GovernmentID"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:GovernmentID" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
GovernmentID	element	GovernmentID	optional / multiple	A collection of government ID records.

15.2.7 Membership

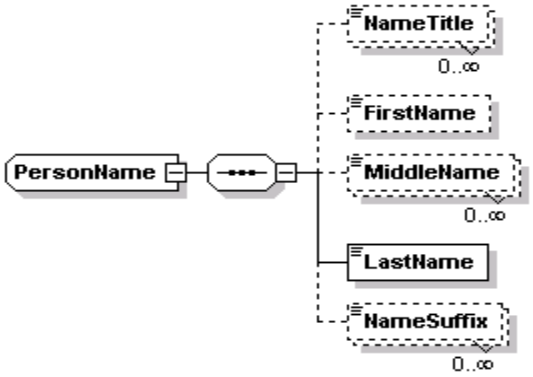


Membership

```
<xs:complexType name="Membership">
    <xs:sequence>
        <xs:element minOccurs="0" name="MembershipType" type="xs:string" />
        <xs:element name="MembershipNumber" type="xs:string" />
        <xs:element minOccurs="0" name="MembershipLevel" type="xs:string" />
        <xs:element minOccurs="0" name="MemberName" type="xs:string" />
        <xs:element minOccurs="0" name="EffectiveDate" type="xs:date" />
        <xs:element minOccurs="0" name="ExpirationDate" type="xs:date" />
        <xs:element minOccurs="0" name="CurrentPoints" type="xs:long" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
MembershipType	element	string	optional	Membership type.
MembershipNumber	element	string	required	Membership number.
MembershipLevel	element	string	optional	Membership level (e.g. GOLD, PLATINUM).
MemberName	element	string	optional	Member name.
EffectiveDate	element	date	optional	Membership effective date (or start date).
ExpirationDate	element	date	optional	Membership expiration date.
CurrentPoints	element	long	optional	Membership point accumulated.

15.2.8 PersonName

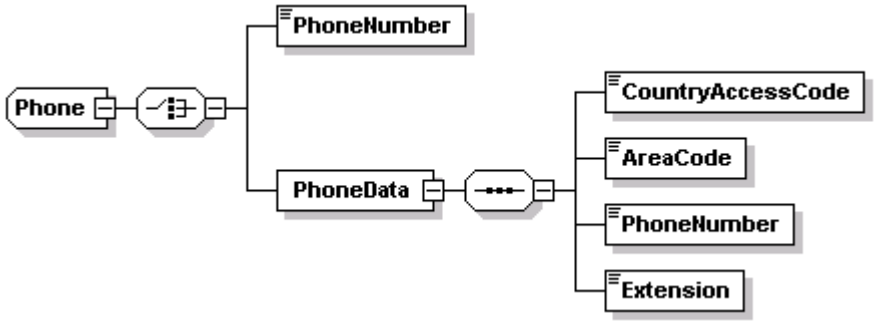


PersonName

```
<xs:complexType name="PersonName">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="NameTitle" type="xs:string" />
        <xs:element minOccurs="0" name="FirstName" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="MiddleName" type="xs:string" />
        <xs:element name="LastName" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="NameSuffix" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="nameOrdered" type="xs:string" />
    <xs:attribute name="familiarName" type="xs:string" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
nameOrdered	attribute	string	none	Not used.
familiarName	attribute	string	none	Familiar name.
NameTitle	element	string	optional / multiple	Name title (e.g. Mr., Mrs., Dr.)
FirstName	element	string	optional	First name.
MiddleName	element	string	optional / multiple	Middle name.
LastName	element	string	required	Last name.
NameSuffix	element	string	optional / multiple	Name suffix (e.g. Jr., III, Esq.)

15.2.9 Phone



Phone

```
<xs:complexType name="Phone">
    <xs:choice>
        <xs:element name="PhoneNumber" type="xs:string" />
        <xs:element name="PhoneData">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="CountryAccessCode" type="xs:string" />
                    <xs:element name="AreaCode" type="xs:string" />
                    <xs:element name="PhoneNumber" type="xs:string" />
                    <xs:element name="Extension" type="xs:string" />
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:choice>
    <xs:attribute name="phoneType" type="xs:string" />
    <xs:attribute name="phoneRole" type="xs:string" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
phoneType	attribute	string	none	Type of phone record. Typical values are HOME or BUSINESS.
phoneRole	attribute	string	none	Phone role. Typical values are PHONE, EMAIL, FAX, WEBPAGE.
PhoneNumber	element	string	required	Phone number, email address, web address or other value appropriate to the phone role attribute.
PhoneData	element	PhoneData	required	The phone number may be specified as either a string above, or as individual components as described in the element below

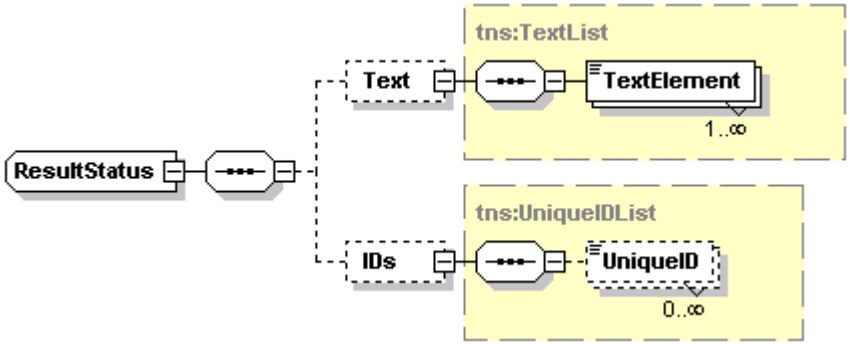
15.2.10 PhoneData

PhoneData

```
<xs:element name="PhoneData">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="CountryAccessCode" type="xs:string" />
            <xs:element name="AreaCode" type="xs:string" />
            <xs:element name="PhoneNumber" type="xs:string" />
            <xs:element name="Extension" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
CountryAccessCode	element	string	required	Country access code.
AreaCode	element	string	required	Area code.
PhoneNumber	element	string	required	Phone number.
PhoneData	element	Extension	required	Phone extension.

15.2.11 ResultStatus



ResultStatus

```
<xs:complexType name="ResultStatus">
    <xs:sequence>
        <xs:element minOccurs="0" name="Text" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:TextList" />
        <xs:element minOccurs="0" name="IDs" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q2:UniqueIDList"
            />
    </xs:sequence>
    <xs:attribute name="resultStatusFlag" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q3:ResultStatusFlag" />
    <xs:attribute name="code" type="xs:string" use="optional" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
resultStatusFlag	attribute	ResultStatusFlag	None	

code	attribute	string	Optional	
Text	element	TextList	Optional	
IDs	element	UniqueIDList	Optional	

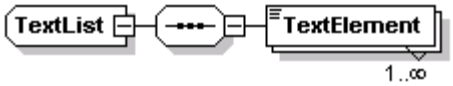
15.2.12 Text

Text

```
<xs:complexType name="Text">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="language" type="xs:language" use="optional" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Text	extension	string	Optional	Text string.
language	attribute	language	Optional	Language code for Text value.

15.2.13 TextList

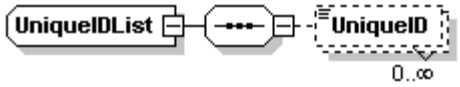


TextList

```
<xs:complexType name="TextList">
    <xs:sequence>
        <xs:element maxOccurs="unbounded" name="TextElement" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q1:Text" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
TextElement	element	Text	required / multiple	

15.2.14 UniqueIDList



UniqueIDList

```
<xs:complexType name="UniqueIDList">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="UniqueID"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueID" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
UniqueID	element	UniqueID	optional / multiple	A list of ID values (see below). Records are often identified by a pair of ID's where the source attribute identifies who each ID belongs to.

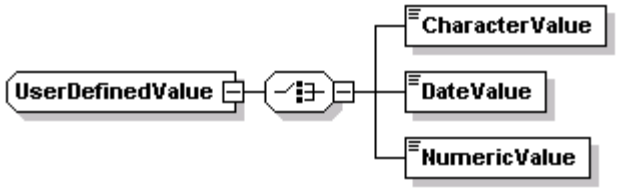
15.2.15 UniqueID

UniqueID

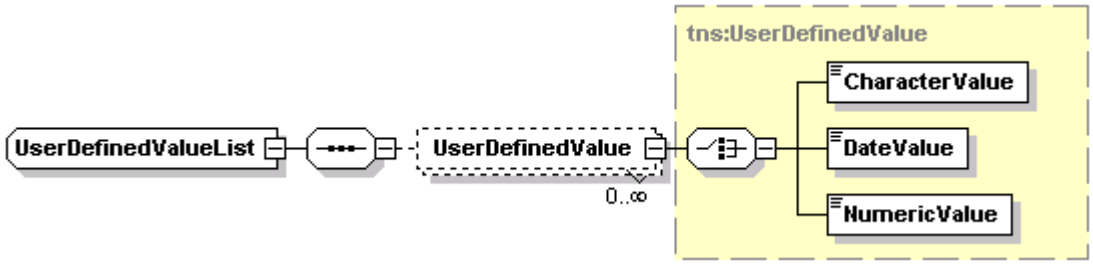
```
<xs:complexType name="UniqueID">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="source" type="xs:string" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
UniqueID	extension	string		The ID value
Source	attribute	string	none	The source for the ID. This value identifies the creator/owner of the ID.

15.2.16 UserDefinedValue



15.2.17 UserDefinedValueList



15.2.18 Simple Types

RecordAdministratorAttributes	Attribute Group	insertUser	Internal user code representing the record creator.
		insertDate	Timestamp record was originally created.
		updateUser	Internal user code representing the user who last updated the record
		updateDate	Timestamp when the record was last updated.
		inactiveDate	Timestamp when the record was inactivated. When null, the record is considered active. When set, the record can be treated as though deleted.
BlackListFlag	Simple Type	REMOVE SET	Enumeration of blacklist values.
Gender	Simple Type	FEMALE MALE UNKNOWN	Enumeration for gender.
ReservationStatusType	Simple Type	CANCELLED CHECKED_IN CHECKED_OUT RESERVED WAITLISTED OTHER REVERSE_CHECKED_IN REVERSE_CHECKED_OUT	Enumeration for reservation statuses.
ResultStatusFlag	Simple Type	FAIL SUCCESS	Enumeration for status results.
SubscriptionAction	Simple Type	OTHER	Enumeration of subscription actions

		SUBSCRIBE UNSUBSCRIBE	to perform.

15.3 Name Schema Elements (name.xsd)

Namespace	http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types

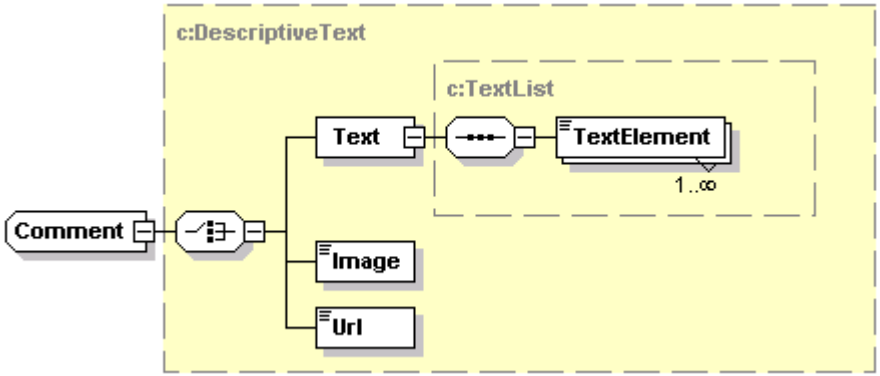
15.3.1 BlackList

BlackList

```
<xs:complexType name="BlackList">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="flag" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:BlackListFlag" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
BlackList	extension	string		Blacklist flag.
Flag	attribute	BlackListFlag	none	One of *REMOVE* or *SET* indicating whether the blacklist flag should be set or not.

15.3.2 Comment



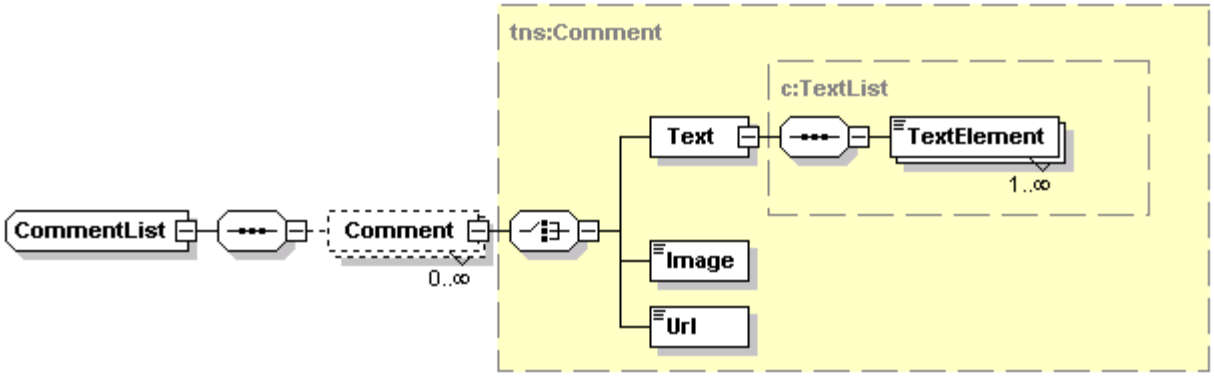
Comment

```
<xs:complexType name="Comment">
    <xs:complexContent mixed="false">
```

```
        <xs:extension xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" base="q1:DescriptiveText">
            <xs:attribute name="commentType" type="xs:string" />
            <xs:attributeGroup ref="q1:RecordAdministratorAttributes" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Comment	extension	DescriptiveText		Extension of DescriptiveText
commentType	attribute	string	none	Type of comment.
RecordAdministratorAttributes	attr group			Record creation and last update attributes.

15.3.3 CommentList

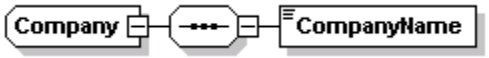


CommentList

```
<xs:complexType name="CommentList">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Comment"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:Comment" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Comment	element	Comment	optional / multiple	A collection of Comment elements.

15.3.4 Company

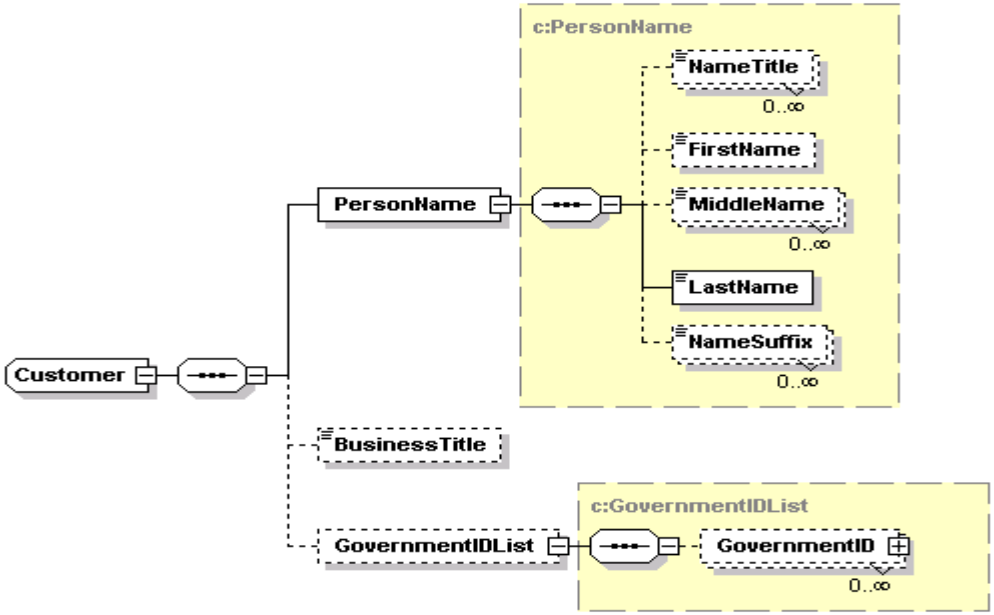


Company

```
<xs:complexType name="Company">
    <xs:sequence>
        <xs:element name="CompanyName" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="commissionCode" type="xs:string" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
commissionCode	attribute	string	none	Not used.
CompanyName	element	string	required	Company name.

15.3.5 Customer



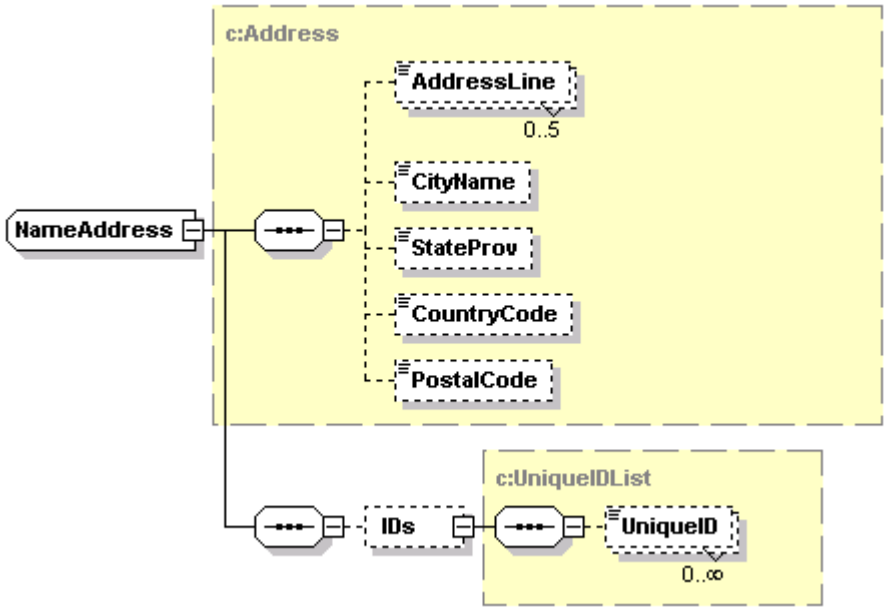
Customer

```
<xs:complexType name="Customer">
    <xs:sequence>
        <xs:element name="PersonName" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:PersonName" />
        <xs:element minOccurs="0" name="BusinessTitle" type="xs:string" />
        <xs:element minOccurs="0" name="GovernmentIDList" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
```

```
            type="q2:GovernmentIDList" />
    </xs:sequence>
    <xs:attribute name="gender" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q3:Gender" />
    <xs:attribute name="birthDate" type="xs:date" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Gender	attribute	Gender	none	The guest's gender.
birthDate	attribute	date	none	The guest's birth date.
PersonName	element	PersonName	required	The guest name record.
BusinessTitle	element	string	optional	The guest's title.
GovernmentIDList	element	GovernmentIDList	optional	A collection of government ID's.

15.3.6 NameAddress



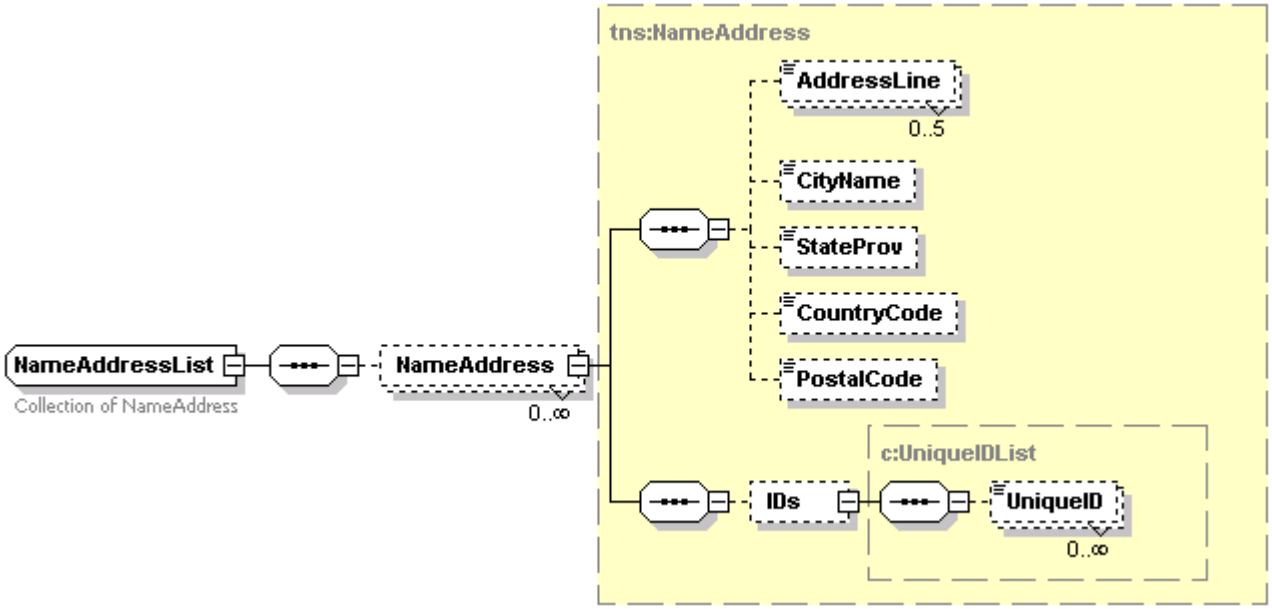
NameAddress

```
<xs:complexType name="NameAddress">
    <xs:complexContent mixed="false">
        <xs:extension xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" base="q1:Address">
```

```
            <xs:sequence>
                <xs:element minOccurs="0" name="IDs" type="q1:UniqueIDList" />
            </xs:sequence>
            <xs:attribute name="primary" type="xs:boolean" />
            <xs:attribute name="displaySequence" type="xs:int" />
            <xs:attributeGroup ref="q1:RecordAdministratorAttributes" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
NameAddress	extension	Address		Extension of Address element type.
Primary	attribute	boolean	none	Flag to indicate primary address.
displaySequence	attribute	int	none	Display sequence.
RecordAdministratorAttributes	attr group			Record creation and last update attributes.
IDs	element	UniqueIDList	optional	Internal ID values.

15.3.7 NameAddressList

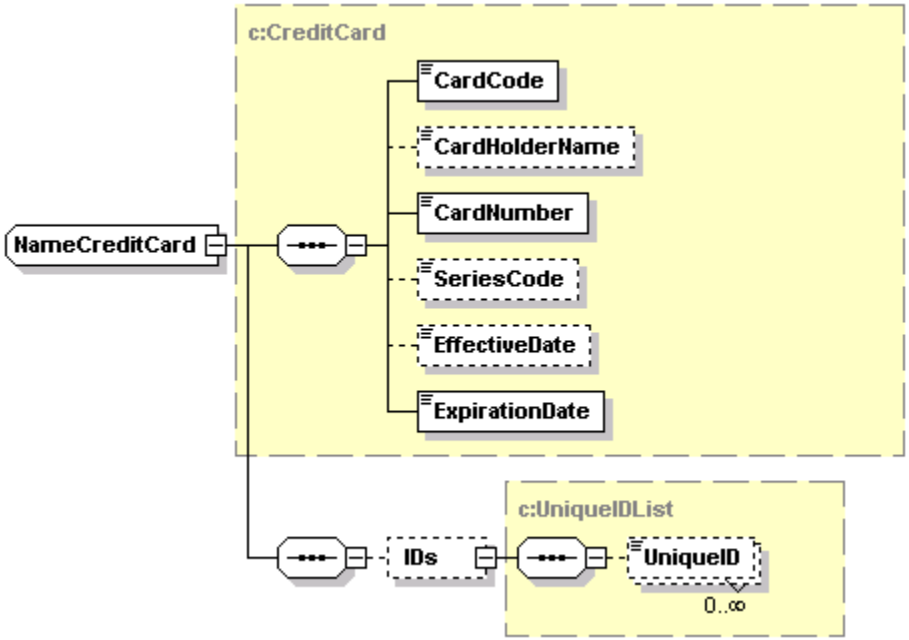


NameAddressList
<xs:complexType name="NameAddressList">

```
    <xs:annotation>
    <xs:documentation>Collection of NameAddress</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="NameAddress"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:NameAddress" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
NameAddress	element	NameAddress	optional / multiple	A collection of NameAddress records.

15.3.8 NameCreditCard



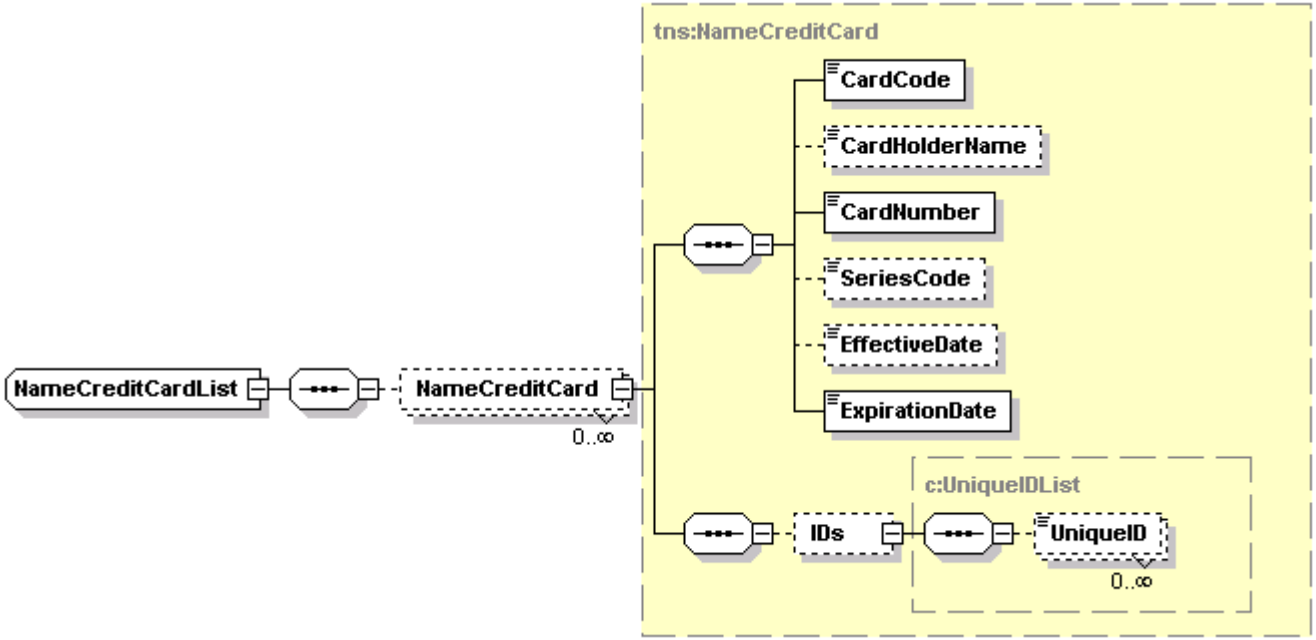
NameCreditCard

```
<xs:complexType name="NameCreditCard">
    <xs:complexContent mixed="false">
        <xs:extension xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" base="q1:CreditCard">
            <xs:sequence>
```

```
            <xs:element minOccurs="0" name="IDs" type="q1:UniqueIDList" />
        </xs:sequence>
        <xs:attribute name="primary" type="xs:boolean" />
        <xs:attribute name="displaySequence" type="xs:int" />
        <xs:attributeGroup ref="q1:RecordAdministratorAttributes" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
NameCreditCard	extension	CreditCard		Extension of CreditCard element.
primary	attribute	boolean	none	Flag to indicate primary credit card.
displaySequence	attribute	int	none	Display order.
RecordAdministratorAttributes	attr group			Record creation and last update attributes.
IDs	element	UniqueIDList	optional	Internal ID's.

15.3.9 NameCreditCardList

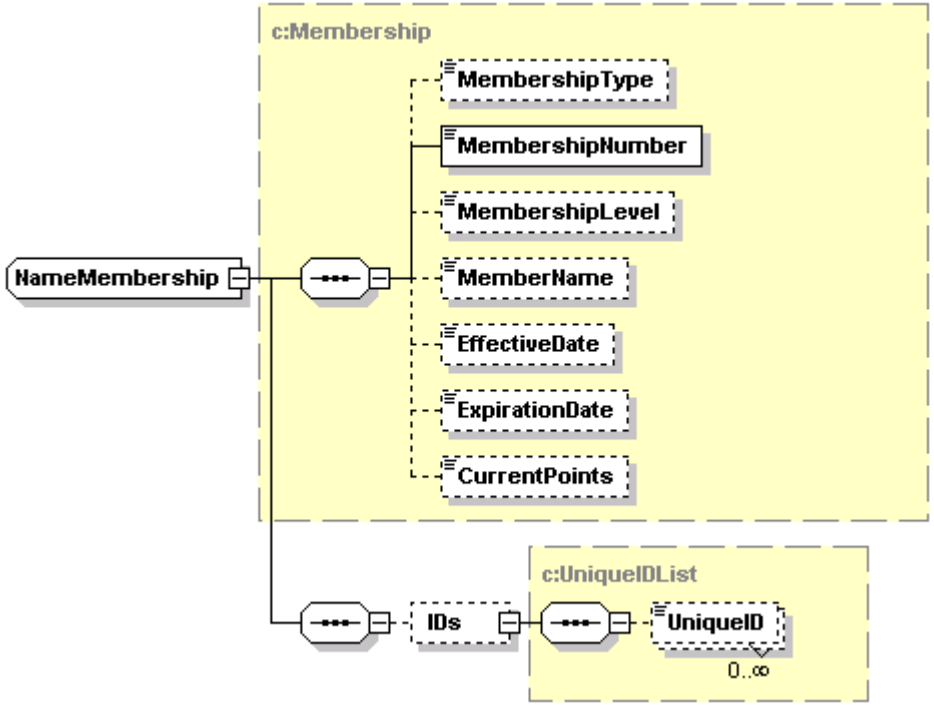


NameCreditCardList
<xs:complexType name="NameCreditCardList">

```
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="NameCreditCard"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:NameCreditCard" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
NameCreditCard	element	NameCreditCard	optional / multiple	A collection of credit card elements.

15.3.10 NameMembership



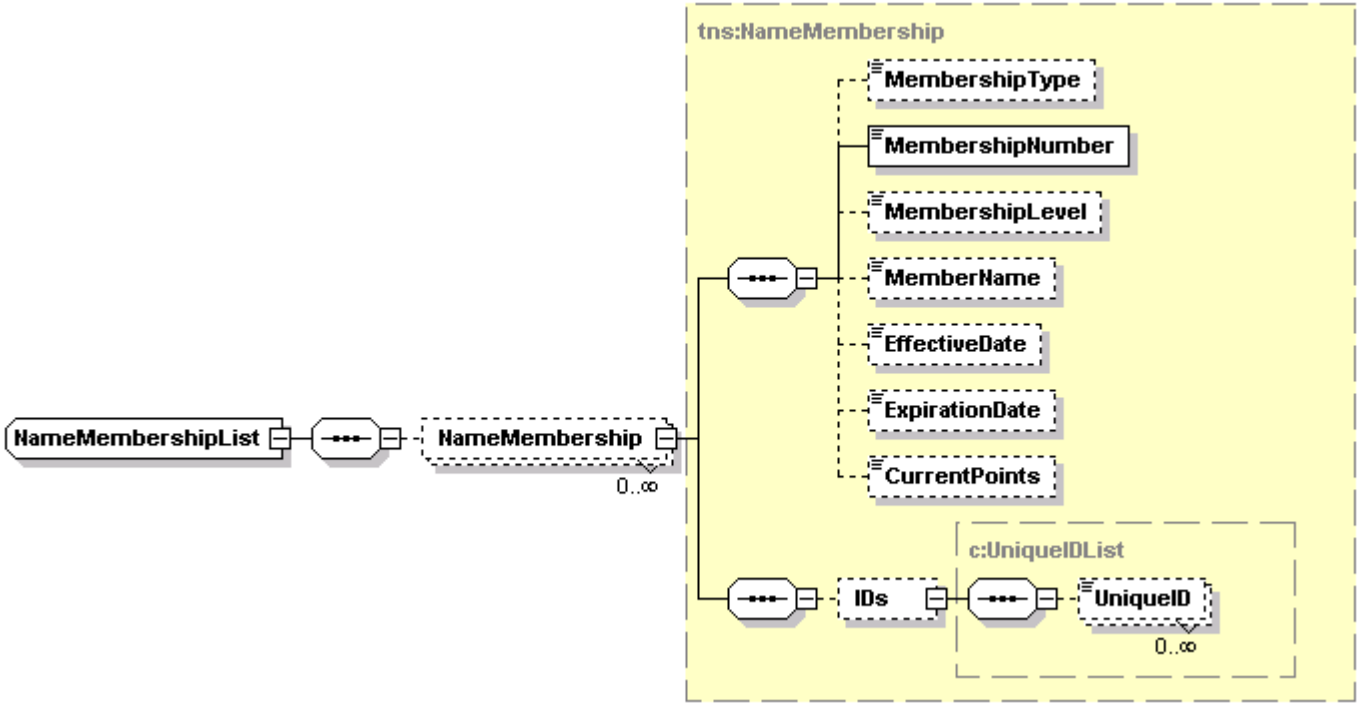
NameMembership

```
<xs:complexType name="NameMembership">
    <xs:complexContent mixed="false">
        <xs:extension xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" base="q1:Membership">
            <xs:sequence>
                <xs:element minOccurs="0" name="IDs" type="q1:UniqueIDList" />
```

```
            </xs:sequence>
            <xs:attribute name="primary" type="xs:boolean" />
            <xs:attribute name="displaySequence" type="xs:int" />
            <xs:attributeGroup ref="q1:RecordAdministratorAttributes" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
NameMembership	extension	Membership		Extension of Membership element.
primary	attribute	boolean	none	Flag to indicate primary membership.
displaySequence	attribute	int	none	Display sequence.
RecordAdministratorAttributes	attr group			Record creation and last update attributes.
IDs	element	UniqueIDList	optional	Internal record ID's.

15.3.11 NameMembershipList



NameMembershipList

```
<xs:complexType name="NameMembershipList">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="NameMembership"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:NameMembership" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
NameMembership	element	NameMembership	optional / multiple	A collection of NameMembership elements.

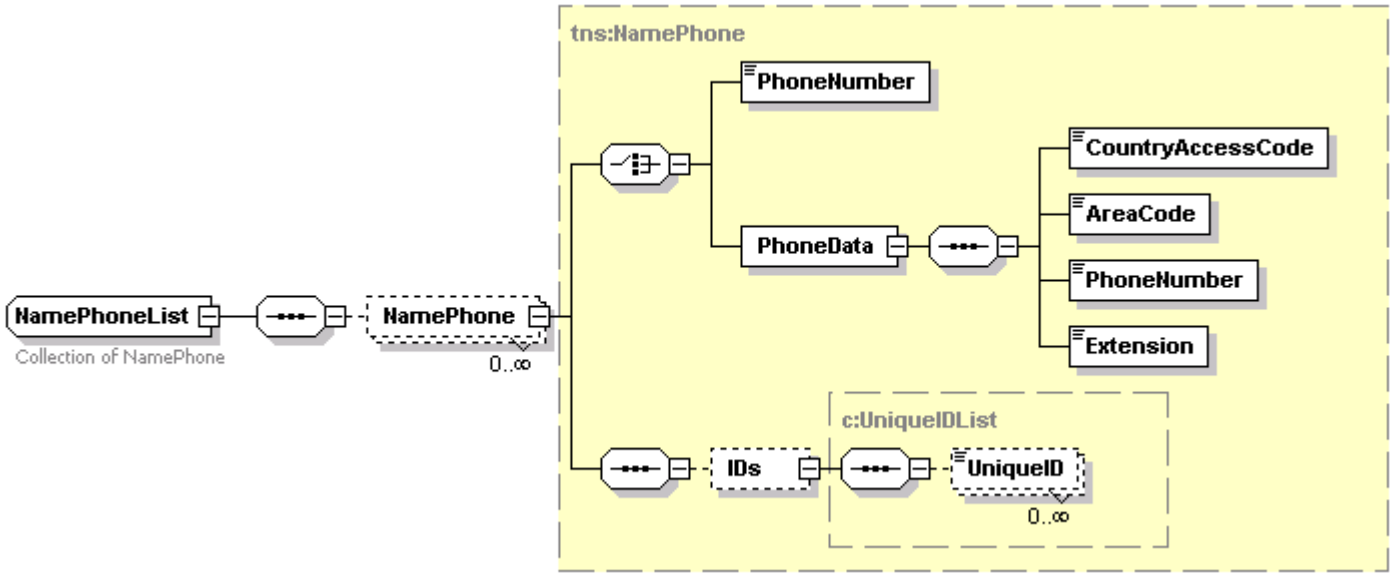
15.3.12 NamePhone

NamePhone

```
<xs:complexType name="NamePhone">
    <xs:complexContent mixed="false">
        <xs:extension xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" base="q1:Phone">
            <xs:sequence>
                <xs:element minOccurs="0" name="IDs" type="q1:UniqueIDList" />
            </xs:sequence>
            <xs:attribute name="primary" type="xs:boolean" />
            <xs:attribute name="displaySequence" type="xs:int" />
            <xs:attributeGroup ref="q1:RecordAdministratorAttributes" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
NamePhone	extension	Phone		Extension of Phone element type.
primary	attribute	boolean	none	Flag to indicate primary phone.
displaySequence	attribute	int	none	Display sequence.
RecordAdministratorAttributes	attr group			Record creation and update attributes.
IDs	element	UniqueIDList	optional	Internal record ID's.

15.3.13 NamePhoneList

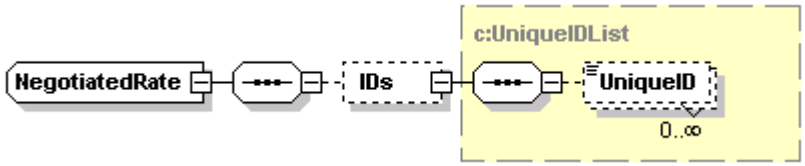


NamePhoneList

```
<xs:complexType name="NamePhoneList">
    <xs:annotation>
    <xs:documentation>Collection of NamePhone</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="NamePhone"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:NamePhone" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
NamePhone	element	NamePhone	optional / multiple	A collection of NamePhone records.

15.3.14 NegotiatedRate

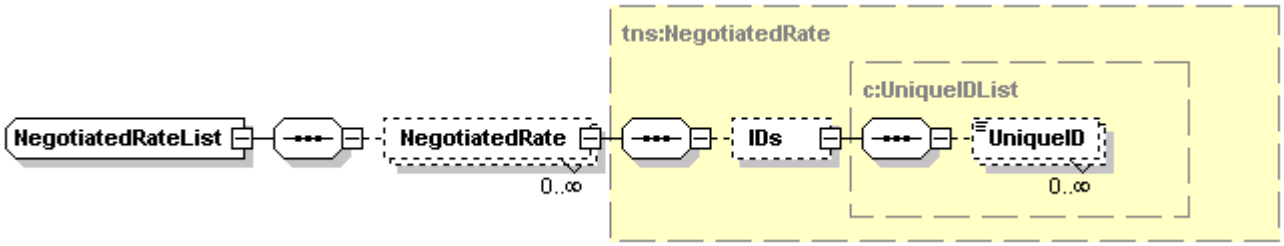


NegotiatedRate

```
<xs:complexType name="NegotiatedRate">
    <xs:sequence>
        <xs:element minOccurs="0" name="IDs" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueIDList"
            />
    </xs:sequence>
    <xs:attribute name="resortCode" type="xs:string" />
    <xs:attribute name="rateCode" type="xs:string" use="required" />
    <xs:attribute name="commissionCode" type="xs:string" />
    <xs:attribute name="beginDate" type="xs:date" />
    <xs:attribute name="endDate" type="xs:date" />
    <xs:attribute name="displaySequence" type="xs:int" />
    <xs:attributeGroup xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" ref="q2:RecordAdministratorAttributes" />
</xs:complexType>
```

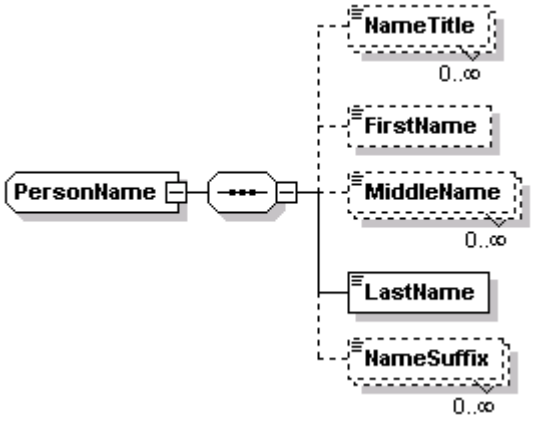
Name	Type	Data Type	Use	Comments
resortCode	attribute	string	none	Property code where negotiated rate is applicable.
rateCode	attribute	string	required	Rate code.
commissionCode	attribute	string	none	Commission code.
beginDate	attribute	date	none	Begin date for rate.
endDate	attribute	date	none	End date for rate.
displaySequence	attribute	int	none	Display sequence.
RecordAdministratorAttributes	attr group			Record creation and last update attributes.
IDs	element	UniqueIDList	optional	Internal record ID's.

15.3.15 NegotiatedRateList



NegotiatedRateList				
<xs:complexType name="NegotiatedRateList">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="unbounded" name="NegotiatedRate"
xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:NegotiatedRate" />
</xs:sequence>
</xs:complexType>				
Name	**Type**	**Data Type**	**Use**	**Comments**
NegotiatedRate	element	NegotiatedRate	optional / multiple	A collection of NegotiatedRate elements.

15.3.16 PersonName

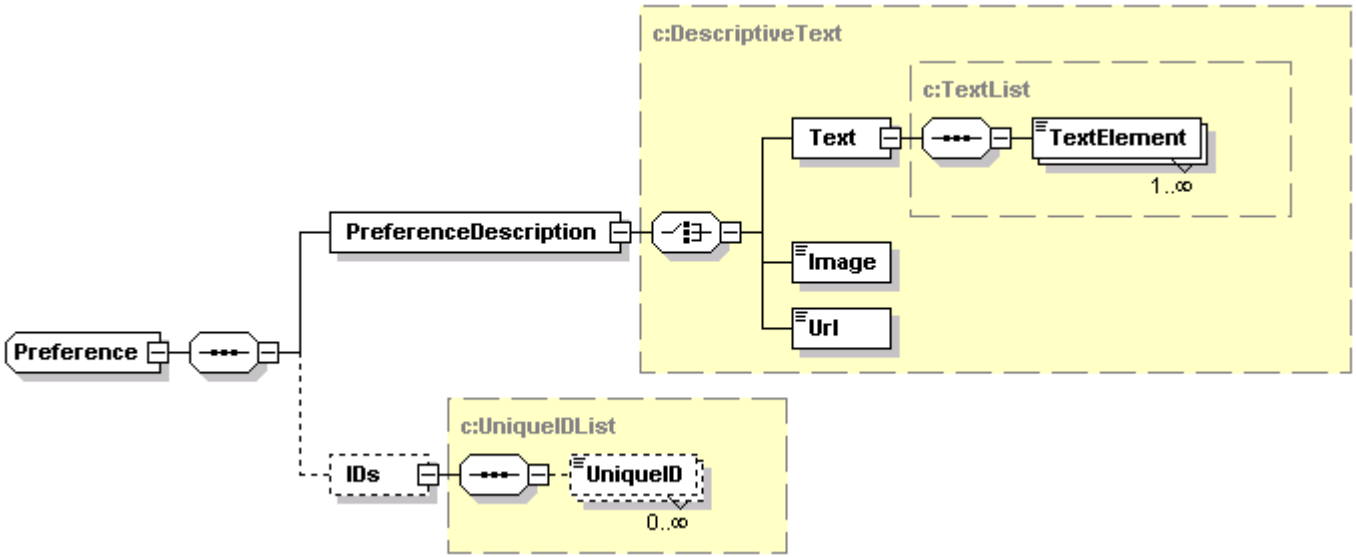


PersonName

```
<xs:complexType name="PersonName">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="NameTitle" type="xs:string" />
        <xs:element minOccurs="0" name="FirstName" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="MiddleName" type="xs:string" />
        <xs:element name="LastName" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="NameSuffix" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="nameOrdered" type="xs:string" />
    <xs:attribute name="familiarName" type="xs:string" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
nameOrdered	attribute	string	none	The ordered representation of the guest name.
familiarName	attribute	string	none	The guest's familiar name.
NameTitle	element	string	optional / multiple	Name title such as "Mr.", "Mrs.", or "Dr."
FirstName	element	string	optional	The guest's first name.
MiddleName	element	string	optional / multiple	The guest's middle name(s).
LastName	element	string	required	The guest's last name. This is the only field which is required.
NameSuffix	element	string	optional / multiple	Name suffix values such as "Jr.", "III", "Esq."

15.3.17 Preference



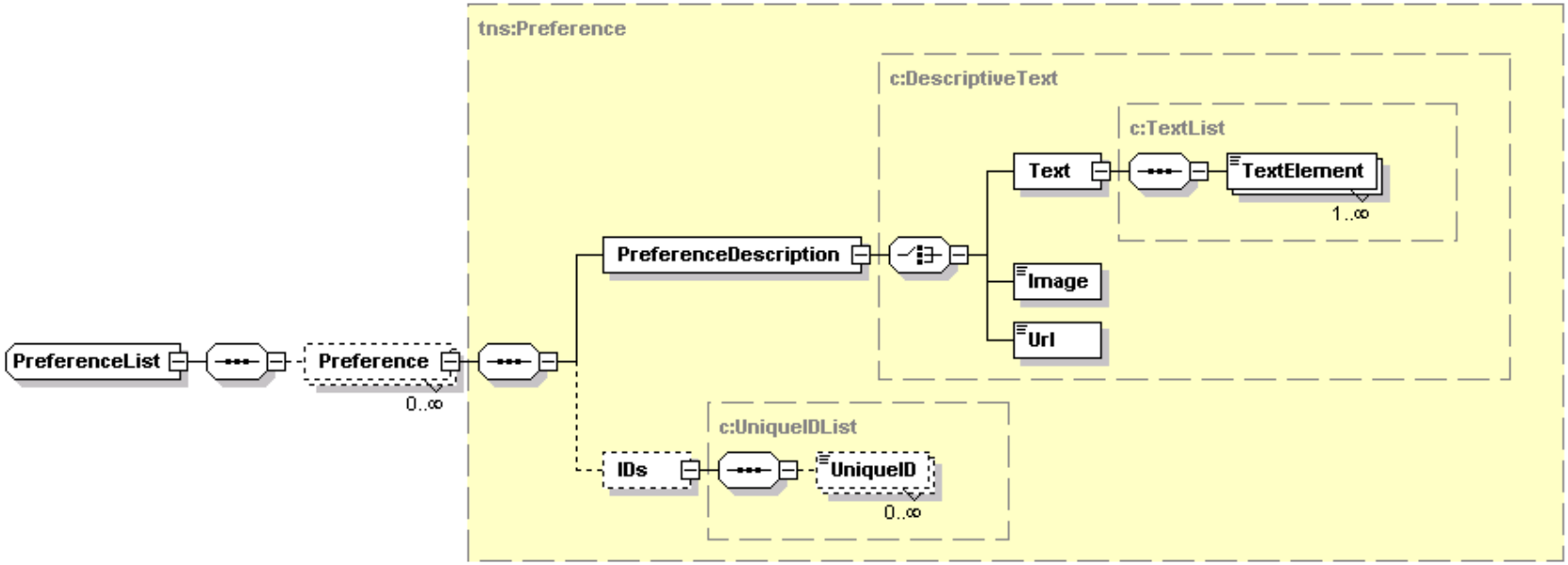
Preference

```
<xs:complexType name="Preference">
    <xs:sequence>
        <xs:element name="PreferenceDescription" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q1:DescriptiveText" />
        <xs:element minOccurs="0" name="IDs" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q2:UniqueIDList"
            />
    </xs:sequence>
    <xs:attribute name="resortCode" type="xs:string" />
    <xs:attribute name="preferenceType" type="xs:string" />
    <xs:attribute name="otherPreferenceType" type="xs:string" />
    <xs:attribute name="preferenceValue" type="xs:string" />
    <xs:attributeGroup xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" ref="q3:RecordAdministratorAttributes" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
resortCode	attribute	string	none	Property code where preference is defined.
preferenceType	attribute	string	none	Preference type code.
otherPreferenceType	attribute	string	none	Not used.
preferenceValue	attribute	string	none	Preference value.
RecordAdministratorAttributes	attr group			Record creation and last update attributes.
PreferenceDescription	element	DescriptiveText	required	Description of the preference.

IDs	element	UniqueIDList	optional	Internal record ID's.

15.3.18 PreferenceList

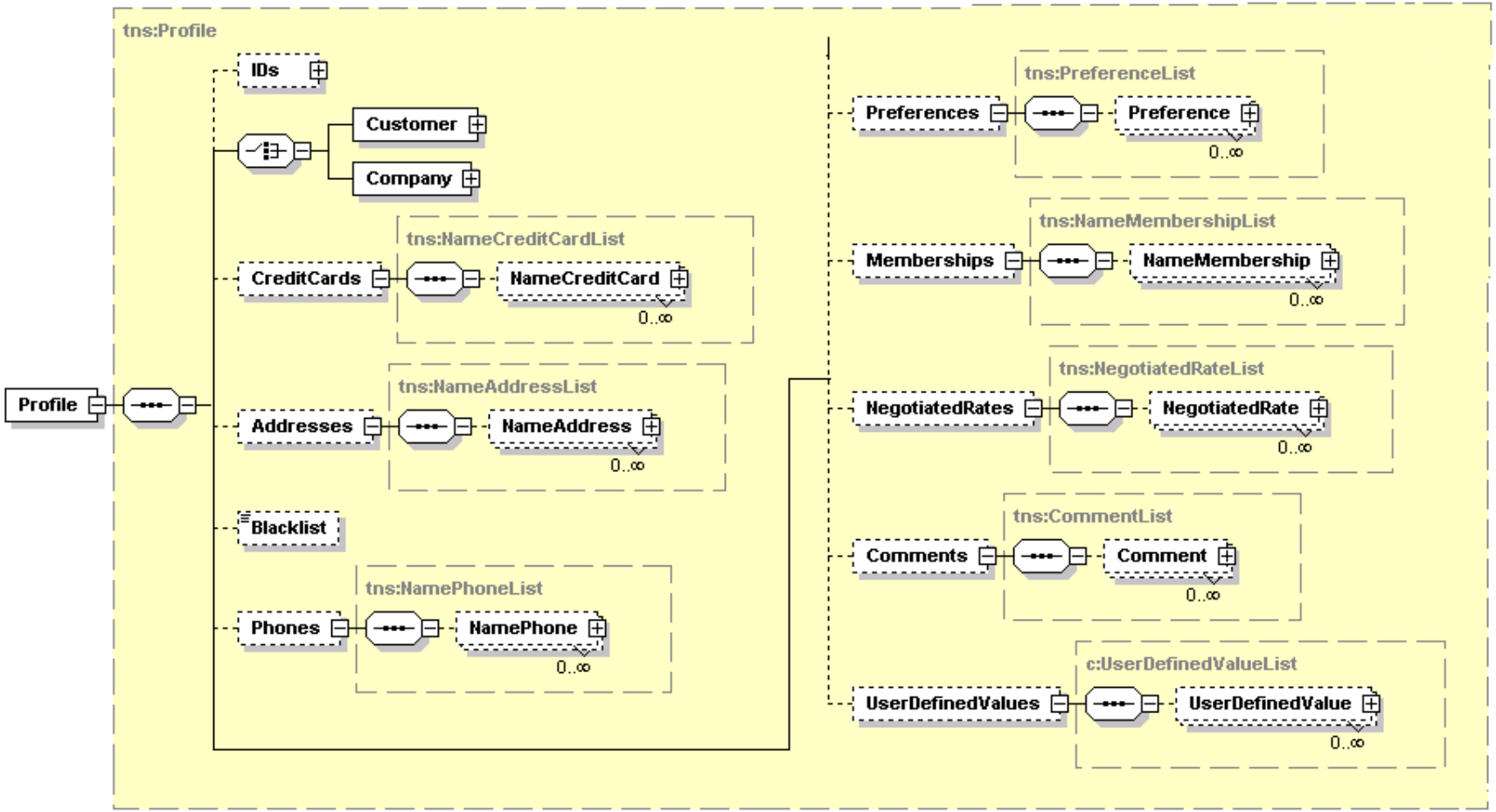


PreferenceList

```
<xs:complexType name="PreferenceList">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Preference"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:Preference" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Preference	element	Preference	optional / multiple	A collection of Preference elements.

15.3.19 Profile



Profile

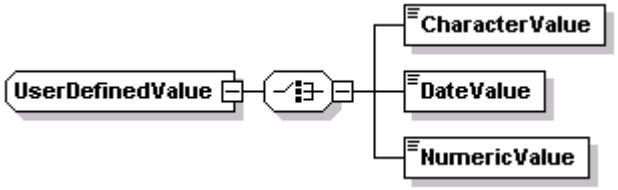
```
<xs:complexType name="Profile">
    <xs:sequence>
        <xs:element minOccurs="0" name="IDs" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueIDList"
            />
        <xs:choice>
            <xs:element name="Customer" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q2:Customer" />
```

```
            <xs:element name="Company" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q3:Company" />
        </xs:choice>
        <xs:element minOccurs="0" name="CreditCards" xmlns:q4="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
            type="q4:NameCreditCardList" />
        <xs:element minOccurs="0" name="Addresses" xmlns:q5="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
            type="q5:NameAddressList" />
        <xs:element minOccurs="0" name="Blacklist" xmlns:q6="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q6:BlackList" />
        <xs:element minOccurs="0" name="Phones" xmlns:q7="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
            type="q7:NamePhoneList" />
        <xs:element minOccurs="0" name="Preferences" xmlns:q8="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
            type="q8:PreferenceList" />
        <xs:element minOccurs="0" name="Memberships" xmlns:q9="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
            type="q9:NameMembershipList" />
        <xs:element minOccurs="0" name="NegotiatedRates" xmlns:q10="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
            type="q10:NegotiatedRateList" />
        <xs:element minOccurs="0" name="Comments" xmlns:q11="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
            type="q11:CommentList" />
        <xs:element minOccurs="0" name="UserDefinedValues" xmlns:q12="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q12:UserDefinedValueList" />
    </xs:sequence>
    <xs:attribute name="nameType" type="xs:string" />
    <xs:attribute name="protected" type="xs:boolean" />
    <xs:attribute name="languageCode" type="xs:string" />
    <xs:attribute name="nationality" type="xs:string" />
    <xs:attribute name="vipCode" type="xs:string" />
    <xs:attribute name="taxExempt" type="xs:boolean" />
    <xs:attributeGroup xmlns:q13="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" ref="q13:RecordAdministratorAttributes" />
</xs:complexType>
```

Name	**Type**	**Data Type**	**Use**	**Comments**
nameType	attribute	string	none	Name type. Typical values are *GUEST* or *COMPANY*.
protected	attribute	boolean	none	Flag to indicated whether profile can be modified by external system.
languageCode	attribute	string	none	Default language code of guest.
nationality	attribute	string	none	Nationality of guest.
vipCode	attribute	string	none	VIP value.
taxExempt	attribute	boolean	none	Not used.
RecordAdministratorAttributes	attr group			Record of creation and last change.
IDs	element	UniqueIDList	optional	Internal profile ID.
Customer	element	Customer	required	The guest name. Either this element or the Company element must be defined. The minimum requirement is the guest's last name.
Company	element	Company	required	The company name. Either this element of the Customer element must be defined.
CreditCards	element	NameCreditCardList	optional	A collection of credit card elements.
Addresses	element	NameAddressList	optional	A collection of address records.
Blacklist	element	BlackList	optional	A flag to indicate whether the guest has been blacklisted.

Phones	element	NamePhoneList	optional	A collection of telephone elements.
Preferences	element	PreferenceList	optional	A collection of guest preference elements.
Memberships	element	NameMembershipList	optional	A collection of membership elements.
NegotiatedRates	element	NegotiatedRateList	optional	A collection of negotiated rate elements.
Comments	element	CommentList	optional	Comments.
UserDefinedValues	element	UserDefinedValueList	optional	A collection of customizable system defined values.

15.3.20 UserDefinedValue

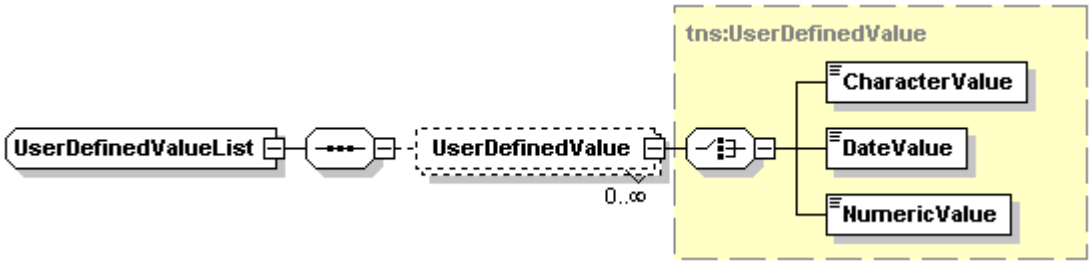


UserDefinedValue

```
<xs:complexType name="UserDefinedValue">
    <xs:choice>
        <xs:element name="CharacterValue" type="xs:string" />
        <xs:element name="DateValue" type="xs:dateTime" />
        <xs:element name="NumericValue" type="xs:float" />
    </xs:choice>
    <xs:attribute name="valueName" type="xs:string" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
valueName	attribute	string	none	User defined value name.
CharacterValue	element	string	required	Character based value (Choice).
DateValue	element	dateTime	required	Date value (Choice).
NumericValue	element	float	required	Numeric value (Choice).

15.3.21 UserDefinedValueList



UserDefinedValueList

```
<xs:complexType name="UserDefinedValueList">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="UserDefinedValue"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UserDefinedValue" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
UserDefinedValue	element	UserDefinedValue	optional / multiple	A collection of user defined values.

Chapter 16 Message Flow

16.1 Overview

The Single Guest Itinerary Team of the Web Service Workgroup within HTNG has created and defined the following information flow concept.

1. Logically link the above defined systems to identify and link Guest Information, Accommodation Reservation and Activity Scheduling data.
2. Allow for Activity Reservation System to retrieve and store updated Accommodation Reservation information, and compare that with the scheduled activities.
3. Allow for Itinerary Consolidation System and Customer Profile System to receive summary Activity schedule information from various Activity Scheduling systems and consolidate into one view of all activities scheduled for a guest.
4. Provide a mechanism to Post charges from the Posting System into the Folio System.
5. Provide a mechanism for other external systems to retrieve consolidated summary Itinerary information

The main key design point is to allow all of the afore mentioned systems to be able to link with each other, using the following unique identifiers:

- Profile ID
- Activity ID
- Reservation ID

A set of messages will be used to maintain the integrity of Guest information between systems. An event model is active between systems to maintain the integrity of the data. Reservations and cancellations are distributed via notification messages.
Interface design was based on a more traditional approach/implementation style of a typical hotel environment, where PMS is playing the roles of ICS, FOL, CRM and ARS, and an Activity System is playing a role of ATS and POS. However, this does not preclude this interface being implemented with different roles assignments.

When vendors go to certify system implementation with each other, they would first have to clearly define and agree which roles will be played by which system in their environment.

16.2 Definitions

16.2.1 Acommodation Reservation System (ARS)

An Accommodation Reservation System is a computerized system managing information about rooms, reservations and customers.

It provides functionality for offering hotel availability, for making accommodation rate and room type inquiries, accommodation reservations, changing, canceling, confirming and viewing reservations, and modifying accommodation rates and types.

The system stores all of the information about the reservation such as customer name(s), address, phone number, number of people/rooms, number of days they will stay, check out date, rate, payment type, etc.

16.2.2 Posting System (POS)

A Posting system is an electronic system that accepts financial data at or near a selling location and transmits that data securely to a computer/system or authorization network for reporting activity, authorization and transaction logging.

A common example of a Posting system is a Point-of-Sale System.

16.2.3 Itinerary Consolidation System (ICS)

An Itinerary Consolidation System is a system that compiles summary data for all customer itinerary reservations in one place. All itinerary bookings will be completed in an “Activity Reservation System” and sent to the “Itinerary Consolidation System”. The link between the “Activity Reservation System” and the “Itinerary Consolidation System” is typically linked pairs of guest identification numbers and optionally, linked pairs of guest reservation numbers. Most commonly the "Itinerary Consolidation System" will also be the “Folio System”.

16.2.4 Activity Reservation System (ATS)

An Activity Reservation System is a system that manages guest reservations for any activity such as: Dining Reservations, Spa Reservations, Tee Time Reservations, and Transportation Reservations. Systems that manage accommodation reservations are not considered activity reservation systems. Some Activity Reservation Systems may process payments or post charges to an external system. The link between an external system and an Activity Reservation system is typically a guest record information which uniquely identify that guest.

16.2.5 Folio System (FOL)

A Folio System is a system that compiles customer bills (commonly known in hotel industry as “folios”). Some charges and payments on folios may be posted by users of the Folio system, while other charges and payments may come through a “posting interface” from external system. The link between an external system and a folio system is an internal reservation ID. Most common example of a Folio System in hospitality industry is a Property Management System.

16.2.6 Customer Profile System (CRM)

A Customer Profile System is system that contains information about hotels guests and customers, such as customer name(s), address, phone number, number of people/rooms, language, demographics, past stay and spending history, etc.

16.2.7 Itinerary Display System (IDS)

An Itinerary Display System is a system that will be able to display guest itinerary to the guest but not necessarily store it. A good example of IDS would be a Pay-Per-View system that would allow a guest to view their itinerary on their guestroom TV.

16.3 Message Structure

Some of the messages are defined as “optional” and some are “mandatory”. Therefore, for certification purposes, any vendor that goes to certify their product against a particular role (POS, ICS, FOL, CRM, ARS, ATS, IDS) should be able to process all mandatory messages.

When an optional message is not supported by a system, that system should return result of “SUCCESS” in order for this not to be perceived as an error”

Messages can be classified into a number of categories within this interface. The following table can be used as a guideline for this purpose.

The following chart represents **Web Services Operations that will be provided by these systems**

Message Scope	**WSDL Operation**	**Implementation**						
		ICS	IDS	ATS	CRM	ARS	FOL	POS
Lookup guest / reservation	ReservationLookup					☑		
	FetchReservation					☑		
	ProfileLookup			☑	☑			
	FetchProfile			☑	☑			
Guest profile management	NewProfile			Opt	Opt			
	UpdateProfile			☑	☑			
Activity management	ActivityLookup	☑		☑				
	CreateActivity	☑						
	UpdateActivity	☑						
	CancelActivity	☑						
	FetchActivities	☑		☑				
Notification	OutOfScopeNotification			☑				
	GuestStatusNotification			☑				
Subscription	Subscription			☑	☑			
Post Charges	PostPayment						☑	
Guest Services	LocationNotification					☑		
	GuestMessage					☑		

Opt *denotes Optional Web Services*

The following chart represents Web Service Operations which systems will **initiate these messages**.

Message Scope	**WSDL Operation**	**Implementation**						
		ICS	IDS	ATS	CRM	ARS	FOL	POS
Lookup guest / reservation	ReservationLookup		☑	Opt				
	FetchReservation			Opt				
	ProfileLookup			☑	☑			
	FetchProfile			☑	☑			
Guest profile management	NewProfile			Opt	Opt			
	UpdateProfile			☑	☑			
Activity management	ActivityLookup	☑						
	CreateActivity			☑				
	UpdateActivity			☑				
	CancelActivity			☑				
	FetchActivities	☑	☑					
Notification	OutOfScopeNotification					☑		
	GuestStatusNotification					☑		
Subscription	Subscription			☑	☑			

Post Charges	PostPayment							☑
Guest Services	LocationNotification			Opt				
	GuestMessage			Opt				

Opt *denotes Optional Messages*

16.4 Use Cases

The following use cases have been identified and illustrate how the various operations/messages defined in the specification can be orchestrated to realize a set of business use cases in the context of a specific system. The use cases are organized into groups where each group represents a system role. The other system that is being interacted with in a specific use case is called the "Provider."

16.4.1 Activity Reservation System (ATS)

16.4.1.1 ATS01 – Select Guest

Name	Select Guest
ID	ATS01
Provider	Customer Profile System
Actor	Activity Reservation System User

16.4.1.1.1 Brief Description

This use cases describes the process when the actor is in the process of locating a Guest record in the ATS. The ATS has the ability to search in both its own guest storage and in the Provider's profile storage. Depending on where the guest/profile is located, once the use case is completed, there must be a record of the guest/profile in both systems and a link must have been established between them.

16.4.1.1.2 Basic Flow

1. The use case starts when the actor chooses to locate a guest. The actor enters in a set of search criteria and searches both the ATS and CRM.
2. The ATS can search for a profile using the following methods
   a. By issuing a **ProfileLookup** call to the CRM, e.g. to search for a profile using name or a phone number, etc.
   b. By issuing a **ReservationLookup** call to the CRM.
3. The actor selects a record (that originated from either the ATS or the CRM)
4. [**OPTIONAL:** If a guest is located in the ATS and not in the CRM]
   a. The ATS will issue a **NewProfile** call to the CRM to create the profile in the CRM.
   b. The ATS will issue a **Subscription** call to the CRM to link the newly created profile in the CRM with the existing guest in the ATS.
5. [If a guest is located in the CRM and not in the ATS]

a. The ATS may choose to call **FetchProfile** on the CRM to fetch more detail about the profile, if needed.

b. The ATS will create the record locally.

c. The ATS will issue a **Subscription** call to the CRM to link the newly created guest record in the ATS to the existing profile in the CRM.

6. [If a guest is located in both systems and already linked]

   a. No action is taken

7. [If a guest is located in both systems but not linked]

   a. The ATS will issue a **Subscription** call to the CRM to link the two records.

8. The use case terminates.

16.4.1.1.3 Preconditions

None

16.4.1.1.4 Postconditions

1. A guest exists in the ATS
2. A profile exists in the CRM
3. The guest in the ATS and the corresponding profile in the CRM have been linked.

16.4.1.2 ATS02 – Update a Guest

Name	Update Guest
ID	ATS02
Provider	Customer Profile System
Actor	Activity Reservation System User

16.4.1.2.1 Brief Description

This use case describes the process when the actor is updating an existing guest record in the ATS. In case the guest record is linked, the linked/external system must be notified of the change as well to keep the systems in sync.

16.4.1.2.2 Basic Flow

1. The use case starts when the actor chooses an existing guest to update in the ATS. The actor then enters in some updated information about the guest and saves the information.

2. [If the guest record is linked]
   a. OPTIONAL: In case the ATS needs to verify the information has not changed in the CRM before issuing an update, the ATS can call one of the following methods to refresh information about the guest: **FetchProfile**, **ProfileLookup**, **ReservationLookup**.
   b. The ATS calls the **UpdateProfile** method on the CRM to update the profile with the new information.
3. The use case terminates.

16.4.1.2.3 Preconditions

1. A guest record exists in the ATS
2. A corresponding, linked, profile exists in the CRM

16.4.1.2.4 Postconditions

1. The guest record in the ATS has been updated
2. The guest profile in the CRM has been updated with the same information

16.4.1.3 ATS03 – Create an Activity

Name	Create an Activity
ID	ATS03
Provider	Customer Profile System, Itinerary Consolidation System
Actor	Activity Reservation System User

16.4.1.3.1 Brief Description

This use case describes the process when the actor is creating an activity for an existing guest. In the case where the actor is creating an activity for a guest that is not yet in the ATS, the ATS01 use case can be executed before this use case is run, although it is not required.

16.4.1.3.2 Basic Flow

1. The use case starts when the actor has created an activity reservation for an existing guest in the ATS.
2. [If the guest record is NOT linked]
   a. The ATS will need either link the guest record to an existing guest record in the CRM, which is can be done through a series of calls to **FetchProfile**, **LookupProfile** and **Subscribe**, in the case of an existing matching profile, or in the case there is no match in the CRM; **NewProfile**, **Subscription**.
3. The ATS will issue an **CreateActivity** call to the ICS.

4. The use case terminates.

16.4.1.3.3 Preconditions

None

16.4.1.3.4 Postconditions

1. There is a record of the newly created activity both in the CRM and ATS.
2. The guest record in the ATS and corresponding profile in the CRM are linked.

16.4.1.4 ATS04 – Update an Activity

Name	Update an Activity
ID	ATS04
Provider	Itinerary Consolidation System
Actor	Activity Reservation System User

16.4.1.4.1 Brief Description

This use case describes the process when the actor is updating an existing activity in the ATS where the guest record is linked to a profile in the CRM.

16.4.1.4.2 Basic Flow

1. The use case starts when the actor has updated an existing activity reservation.
2. The ATS will issue an **UpdateActivity** call to the ICS.
3. The use case terminates.

16.4.1.4.3 Preconditions

1. The related guest record in the ATS is linked to a profile in the CRM.

16.4.1.4.4 Postconditions

1. There is a matching record of the updated created activity both in the CRM and ATS.

16.4.1.5 ATS05 – Cancel an Activity

Name	Cancel an Activity
ID	ATS05
Provider	Itinerary Consolidation System

Actor	Activity Reservation System User

16.4.1.5.1 Brief Description

This use case describes the process when the actor is canceling an existing activity in the ATS where the activity is for a linked guest.

16.4.1.5.2 Basic Flow

1. The use case starts when the actor has canceled an existing activity reservation.
2. The ATS will issue a **CancelActivity** call to the ICS.
3. The use case terminates.

16.4.1.5.3 Preconditions

1. The related guest record in the ATS is linked to a profile in the CRM.

16.4.1.5.4 Postconditions

1. The Activity reservation is canceled in both the ATS and the CRM.

16.4.1.6 ATS06 – Guest Message

Name	Guest Message
ID	ATS06
Provider	Accommodation Reservation System
Actor	Activity Reservation System User

16.4.1.6.1 Brief Description

This use case describes the process when the actor is sending a message to a guest. Accommodation Reservation System may generate a guest notification, such as turning on the message light in the guest's room.

16.4.1.6.2 Basic Flow

1. The use case starts when the actor has sent a message to a linked guest.
2. The ATS will issue a **GuestMessage** call to the ARS passing ResevationID
3. The use case terminates.

16.4.1.6.3 Preconditions

1. The related guest record in the ATS is linked to a profile in the CRM which has a reservation in the ARS

16.4.1.6.4 Postconditions

1. The guest message is submitted to the ARS.

16.4.1.7 ATS07 – Guest Location Notification

Name	Guest Location Notification
ID	ATS07
Provider	Accommodation Reservation System
Actor	Activity Reservation System User

16.4.1.7.1 Brief Description

This use case describes the process when a location notification message is submitted by the actor in order to notify the Accommodation Reservation System of a guest's current physical location. ARS may implement the feature of locating a guest in order to forward phone calls or deliver messages.

Basic Flow

1. The use case starts when the actor has sent a location notification message to a linked guest.
2. The ATS will issue a **LocationNotification** call to the ARS passing ResevationID
3. The use case terminates.

16.4.1.7.2 Preconditions

1. The related guest record in the ATS is linked to a profile in the CRM which has a reservation in the ARS

16.4.1.7.3 Postconditions

1. The guest location notification message is submitted to the ARS.

16.4.2 Itinerary Display System (IDS)

16.4.2.1 IDS01 – Show Itinerary

Name	Show Itinerary

ID	IDS01
Provider	Accommodation Reservation System / Itinerary Consolidation System
Actor	Guest

The guest can view his or hers own itinerary by implicitly querying the IDS through some sort of guest User Interface, e.g. a web site, an in-room interactive TV, etc.

16.4.2.1.1 Basic Flow

1. The use case starts when the actor chooses to view his or her own itinerary.
2. The IDS issues a **ReservationLookup** call to the ARS. The search criteria supplied to the **ReservationLookup** call will typically be one of the following identifiers:
   a. A ProfileID if a ProfileID is already known, e.g. because the actor has logged onto a web site and the profile is already known.
   b. A Confirmation Number
   c. A Room Number of an in-house guest.
3. [If more than one reservation/stay was found]
   a. The system asks the actor to choose the reservation/stay for which to retrieve the itinerary for.
   b. The actor selects one of the reservations/stays.
4. The IDS issues a **FetchActivities** call to the ICS to retrieve all the activities for the (now known) ProfileID and ReservationID.
5. The IDS displays the itinerary to the guest.
6. The use case terminates.

16.4.2.1.2 Preconditions

1. There exists at least one reservation in the ARS for the guest in question.

16.4.2.1.3 Postconditions

None

16.4.3 Accommodation Reservation System (ARS)

16.4.3.1 ARS01 Create a Reservation

Name	Create a Reservation
ID	ARS01
Provider	Activity Reservation System
Actor	Typical ARS user

Whenever a reservation is created in the ARS that involves a linked guest/profile, the ARS should be able to notify subscribers of the new reservation. While the ability to generate this notification is required, the activation of the notification should remain configurable, and implementation dependent.

16.4.3.1.1 Basic Flow

1. The use case starts when the actor has changed or created a reservation in ARS.
2. [If the associated guest/profile is linked]
   a. The ARS will issue a **GuestStatusNotification** call to the ATS.
3. The use case terminates.

16.4.3.1.2 Preconditions

1. The guest record is linked.

16.4.3.1.3 Postconditions

1. The reservation is canceled in both the ATS and the CRM.

16.4.3.2 ARS02 –Change in Reservation Status

Name	Change Reservation Status
ID	ARS02
Provider	Activity Reservation System
Actor	Typical ARS user

Whenever a status change is made to a reservation with a linked guest profile in the ARS, the ARS should notify subscribers of the change. For example, when a guest checks in or out, the ATS would typically be notified of the change in guest status.

16.4.3.2.1 Basic Flow

1. The use case starts when the actor checks in or out an existing linked reservation.
2. [If the associated guest/profile is linked]
   a. The ARS will issue a **GuestStatusNotification** call to the ATS.

3. The use case terminates.

16.4.3.2.2 Preconditions

1. The guest record is linked.

16.4.3.2.3 Postconditions

1. None.

16.4.3.3 ARS03 –Cancel or Move a Reservation

Name	Cancel or Move a Reservation
ID	ARS03
Provider	Activity Reservation System
Actor	Typical ARS user

Whenever a reservation which has linked activities is cancelled or moved in the ARS, it may be desirable to notify the ATS of the reservation change. While the ability to generate the notification is required, it should be operationally configurable and implementation dependent.

16.4.3.3.1 Basic Flow

1. The use case starts when the actor has cancels or moves a reservation in ARS which has linked activities.
2. [If the reservation contains linked activities, that ARS may (per actor option or implementation default):]
   a. Issue an **OutOfScopeNotification** message to the ATS to release (cancel) the associated activites.
   b. Issue an **OutOfScopeNotification** message to the ATS to notify the reservation change for later follow up action.
   c. Make no notification.
3. The ATS would perform the following actions per the conditions above.
   a. The activity is automatically cancelled.
   b. An actor in the ATS system follows up to determine whether the activity should be cancelled.
   c. No action occurs in the ATS.
4. The use case terminates.

16.4.3.3.2 Preconditions

1. The reservation contains linked activities..

16.4.3.3.3 Postconditions

1. The reservation is canceled or moved in the ARS. The activities in the ATS may be cancelled.

16.4.4 Itinerary Consolidation System (ICS)

16.4.4.1 ICS01 – Retrieve Itinerary

Name	Retrieve Itinerary
ID	ICS01
Provider	Activity Reservation System
Actor	Typical ICS user

The ICS has the ability to query the ATS for (typically unlinked) guest records and reservations for that guest.

16.4.4.1.1 Basic Flow

1. The use case starts when the actor wishes to view the itinerary for a guest, which is likely unlinked, although the actor may or may not be aware of that. The actor enters in a set of search criteria.
2. The ICS can query the ATS for a profile using either a **ProfileLookup** or **ActivityLookup** call. The system displays the results to the actor (perhaps incorporated with its own matching lists of guests).
3. The actor selects a guest
4. The ICS can query the ATS for more detailed profile information by issuing a **FetchProfile** call, and the ICS issues a **FetchActivities** call to the ATS to fetch all the activities for that guest. The ICS will then issue a **Subscription** call to the ATS to subscribe for changes to the profile that was just brought over.
5. The system displays the itinerary to the actor.
6. The use case terminates.

16.4.4.1.2 Preconditions

None

16.4.4.1.3 Postconditions

1. The guest selected guest record is linked in both systems.

2. The ICS has an up-to-date list of activities for the selected guest.

16.4.5 Customer Profile System (CRM)

16.4.5.1 CRM01 – Select Guest

Name	Select Guest
ID	CRM01
Provider	Activity Reservation System
Actor	Typical Customer Profile System user

16.4.5.1.1 Brief Description

This use cases describes the process when the actor is in the process of locating a Guest record in the CRM. The CRM has the ability to search in both its own guest storage and in external ATS systems. If the profile is retrieved from the external ATS, there must be a record of the guest/profile in both systems and a link must have been established between them.

16.4.5.1.2 Basic Flow

1. The use case starts when the actor chooses to locate a guest. The actor enters in a set of search criteria and searches both the ATS and CRM.
2. The ATS can search for a profile using the following methods
   a. By issuing a **ProfileLookup** call to the ATS, e.g. to search for a profile using name or a phone number, etc.
   b. By issuing a **ActivityLookup** call to the ATS.
3. The actor selects a record (that originated from either the ATS or the CRM)
4. [If a guest is located in the ATS and not the CRM]
   a. The CRM may choose to call **FetchProfile** on the ATS to fetch more detail about the profile, if needed.
   b. The CRM will create the record locally.
   c. The CRM will issue a **Subscription** call to the ATS to link the newly created guest record in the CRM to the existing profile in the ATS.
5. [If a guest is located in both systems and already linked]
   a. No action is taken
6. [If a guest is located in both systems but not linked]

a. The CRM will issue a **Subscription** call to the ATS to link the two records.

7. The use case terminates.

16.4.5.1.3 Preconditions

None

16.4.5.1.4 Postconditions

1. A guest exists in the CRM
2. A profile may exist in the ATS (if retrieved from the ATS)
3. The guest in the CRM and the corresponding profile in the ATS (if exists) have been linked.

16.4.5.2 CRM02 – Update a Guest

Name	Update Guest
ID	CRM02
Provider	Activity Reservation System
Actor	Typical Customer Profile System user

16.4.5.2.1 Brief Description

This use case describes the process when the actor is updating an existing guest record in the CRM. In case the guest record is linked, the linked/external system must be notified of the change as well to keep the systems in sync.

16.4.5.2.2 Basic Flow

1. The use case starts when the actor chooses an existing guest to update in the CRM. The actor then enters in some updated information about the guest and saves the information.
2. [If the guest record is linked]
3. The CRM calls the **UpdateProfile** method on the ATS to update the profile with the new information.
4. The use case terminates.

16.4.5.2.3 Preconditions

1. A guest record exists in the CRM
2. A corresponding, linked, profile exists in the ATS

16.4.5.2.4 Postconditions

1. The guest record in the CRM has been updated
2. The guest profile in the ATS has been updated with the same information

16.4.6 Posting System (POS)

16.4.6.1 POS01 – Posting Guest Related Charges

Name	Posting Guest Related Charges
ID	POS01
Provider	Folio System
Actor	Posting System user

16.4.6.1.1 Brief Description

This use cases describes the process when the actor adds a charge or submits a refund to a guest folio which is managed by the Folio System.

16.4.6.1.2 Basic Flow

1. The use case starts when the actor chooses to submit a guest's related charge or refund to the Folio System.
2. The POS will issue a **PostPayment** (pmtpostingupdate)
3. The POS will receives back a PostPayment response message (pmtpostingresults)
   a. If the response is successful no action is taken
   b. If the response is failure the POS will retry the sequence of PostPayment
4. The use case terminates.

16.4.6.1.3 Preconditions

5. A corresponding guest folio exists in the FOL

16.4.6.1.4 Postconditions

1. A guest related charges and refunds exist in the FOL

Part 3 Digital Signage Version 1.0

Chapter 17 Document Information

17.1 Change History

17.1.1 Messaging Requirements

Date	Description	Author
September 2006	First Draft	Neil Schubert, Angela Skoda
October 6, 2006	Updated based on input from Oct. 3 meeting in Boston	Angela Skoda
October 20, 2006	Updated based on the October 13 and October 20 conference calls	Angela Skoda
November 7, 2006	Removed push of data from the Sales and Catering system. Remove references to a reservation identifier	Angela Skoda
November 30, 2006	Added “Response Complete” flag	Angela Skoda

17.1.2 Messaging Specification

Date	Author	Comments
01/19/2007	Alex Shore, Newmarket International, Inc.	Initial draft
02/02/2007	Alex Shore, Newmarket International, Inc.	Schema changes from Digital Signage conference call. For MeetingSpaceResponse – require @meetingKey and Group/@name. For Characteristics request – require property key. For response, added property node similar to how the MeetingSpaceResponse is organized. Updated sample SOAP xml in appendix.
03/23/2007	Alex Shore, Newmarket International, Inc.	Changes from the 03/16/2007 conference call: Fixed namespace typo for MeetingSpaceRequest in the wsdl. Added new fault type for too large of a date range specified.
4/15/2008	Ken Jones	1. Updated MeetingSpaceResponse schema as defined below. 2. Removed sample XML from this document, and added a reference to the actual sample XML included with the API.
5/15/2008	Ken Jones	Removed Draft label from cover page.

17.1.3 Schema Changes

Date	Author	Schema	Comments
01/11/2007	Alex Shore	All except fault schemas	Version 0.3 schemas posted to the HTNG collaboration site for review by Digital Signage group.
01/19/2007	Alex Shore	All	Changed verbiage to use MeetingSpace and MeetingSpaceCharacteristics instead of Event and MeetingRoom.
02/02/2007	Alex Shore	MeetingSpaceCharacteristicsRequest.xsd, MeetingSpaceCharacteristicsResponse.xsd, MeetingSpaceResponse.xsd	See above Document changes notes for v1.1.
03/23/2007	Alex Shore	MeetingSpaceServiceFaults.xsd	Added fault for too large of a date range.
05/15/2008	Ken Jones	MeetingSpaceResponse.xsd	1. Change Room Key length restrictions from 3 characters to 1 (Element: MeetingSpace). 2. Link Sub Meetings (Sub Functions) to their parent Meeting (Main Function) 3. Add Backup/Overflow Room Key to the MeetingSpaceResponse file (Element: MeetingSpace) and allow multiple occurrences of BackupMeetingRoom and OverFlowRoom elements. 4. Restructure the schema for the MeetingSpaceResponse to better represent the relationship between Event, Meetings and Sub-Meetings and to avoid duplication of the Event information on each Meeting Space. 5. Change the startDate,startTime attributes to a single attribute named startDateTime defined as xsd:DateTime

17.2 Purpose

This document provides functional, business, and technical descriptions of the web service requests that are available for querying Meeting Space information from a Sales & Catering application. Expertise in XML, SOAP, and HTTP is required.

The Meeting Space API is a synchronous request/response architecture that provides real-time data back to the caller. This API uses an HTNG 2.0 (draft)-compliant web service.

17.3 Terminology

For the purpose of this document the following terms have been defined as follows:

Term	Definition
Vendor	For the purposes of this document, a Vendor is any client that uses the Meeting Space Request web service. An example of a Vendor is a digital signage software application.
Meeting Space	A meeting being used by a group at a hotel for a certain period of time.
Event	A booking associated with a company, account, or agency. A booking can have sleeping rooms and meeting space information.
HTNG	The HTNG protocol is fully described at **www.htng.org**. The Meeting Space services are using the HTNG 2.0 framework draft specification.
Property	For the purposes of this document, "property" is the term used for a hotel, facility, resort, etc. The term is used generically.
messageID	GUID that identifies a specific transaction. This value is set by the client in the WS-Addressing SOAP header.
PropertyKey	This value uniquely defines a hotel property for a sales & catering system.

17.4 Referenced Documents

The following table shows the documents on which this API document is dependent:

Name	Description	Location
HTNG 2.1 Framework	Describes the web service framework standards that workgroups must adhere to.	Part 1 of this document
Digital Signage Messaging Requirements	Document made by the Digital Signage workgroup to define the requirements of the meeting space interface.	Included as Chapter 18 of this document

Chapter 18 Digital Signage Messaging Requirements

18.1 Background

This chapter is a collection of Digital Signage Messaging requirements provided by the HTNG Digital Signage Workgroup. Collectively, we have decided to focus the first release of this specification on the messaging between Sales and Catering systems and Digital Signage Vendors. Other sources of information for digital signage, for example, PMS systems, HR systems, Internet Media, etc. will be addressed in future releases.

18.2 Functional Requirements

Hotel digital signage deployment consists of television and/or video monitor display technology deployed in public space areas in order to communicate messages to hotel guests and employees. Examples of digital signage applications include locations such as, lobby walls, restaurants and bars, meeting room entry or foyer, conventional halls, exhibit halls, kiosks, and back of house employee systems. Guestroom applications are also used to present messages on hotel room televisions and, potentially, other display technologies such as display telephones.

While this document focuses on the requirements for sales and catering system messaging, we acknowledge that a more comprehensive messaging standard will be required in future releases of this specification.

18.2.1 Overall Assumptions

1. The current business requirement will only accommodate request/response messages initiated by the digital signage application. Future releases of this specification will address a "push" of data from the sales and catering source system.

2. The digital signage application can pull information from the source system either through an automated batch pull or a user-initiated trigger within the digital signage application.

3. The scope of this interface has been limited to messages between a Sales and Catering system and a digital signage application. Information about an individual guest and their participation in an event would require separate PMS integration, which is currently considered out of scope, but will be considered in future releases of this specification.

4. A full set of information will be passed and it is up to the digital signage application to recognize changes, if needed.

5. Each hotel will need to have the ability to regulate and control the frequency of any automated pull of data initiated from the digital signage application. Performance concerns around the frequency of data transfer must be considered.

6. Each hotel will need to have the ability to regulate the number of transactions that can be passed through the interface.

Definitions:

- Group – An account, for example, IBM.
- Event – An entire piece of group business, for example, IBM North America Conference
- Meeting – A subset of an event, usually in a single room, for example, a breakout session.

18.2.2 Sales and Catering Systems

Selection Criteria – The digital signage application will need the ability to request sales and catering information associated with any combination of the following:

- by property identifier
- by date/time range (with some limit on the number of days)
- by event identifier
- by exhibit flag (Y/N)
- by "postable" flag (Y/N)
- by room grouping (physical location, such as a wing)

This request message should also contain an identifier that specifics how this request was initiated, either through a batch process or through a user-initiated process. This identifier would be appended to the request.

Proposed Sales and Catering Response Data

- Hotel Name and related identifier
- Group Name and related identifier
- Group logo URL or pathname
- Event Name and related identifier
- Event logo URL or pathname
- Event "Post As" Name
- Host Event Identifier (parent of this event, used to indicate a Host/Affiliate relationship)
- Meeting Room Space Display Name and related identifier
- Meeting Name and related identifier (e.g., breakfast, lunch)

- Meeting "Post As" Name (free form description of meeting)
- Date
- Start Time (local time for the property)
- End Time (local time for the property)
- Postable Flag -Y/N
- Exhibit Flag – Y/N
- Backup Meeting Room Space Name
- Overflow Meeting Room Space Name
- Response Complete – Y/N

18.2.3 Meeting Space Characteristics

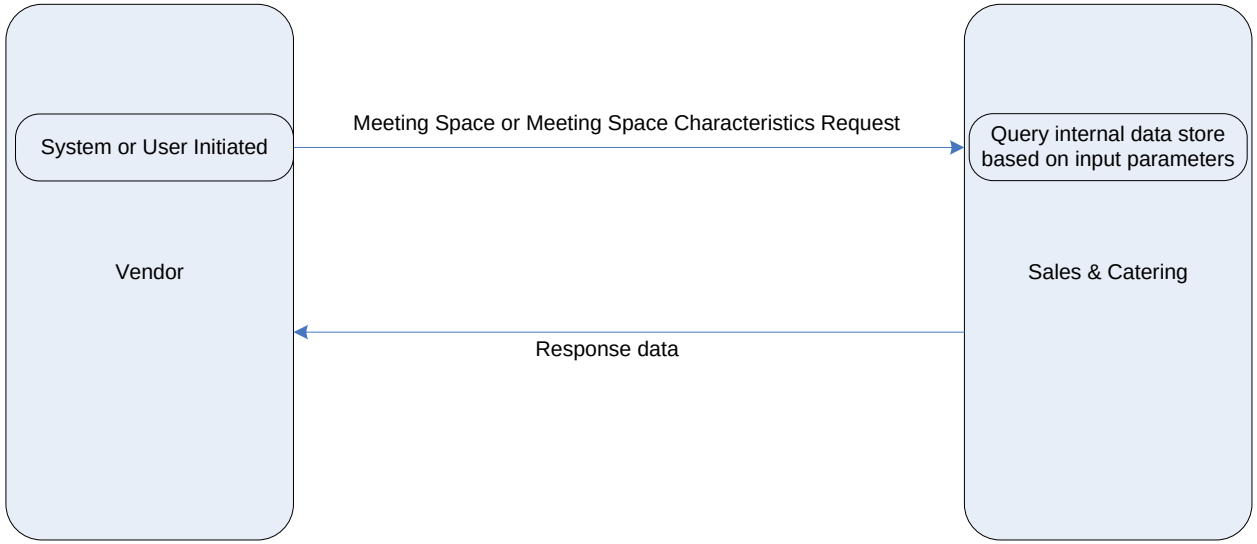
Separate request/response. This data will be refreshed very rarely, when there is a change in meeting room names or configuration.

- Meeting Room Identifier
- Meeting Room Space Display Name
- Floor number
- Floor description
- Room grouping (physical location like a wing, building, etc)
- Directions to the meeting room
- Room Type (Divisible/Indivisible)
- Parent Meeting Room Identifier
- Map reference number

Chapter 19 Functional Overview

19.1 Functional Flow

Below is an example of the flow that would be executed when a Vendor requests Meeting Space or Meeting Space Characteristics data from the Sales & Catering system's web service.



19.2 System Roles

The following table defines the system roles for this interface:

Message	Sales and Catering	Digital Signage	Property Management System
Request Meeting Space	Web Service Provider	Web Service Consumer	Web Service Consumer
Request Meeting Space Characteristics	Web Service Provider	Web Service Consumer	Web Service Consumer

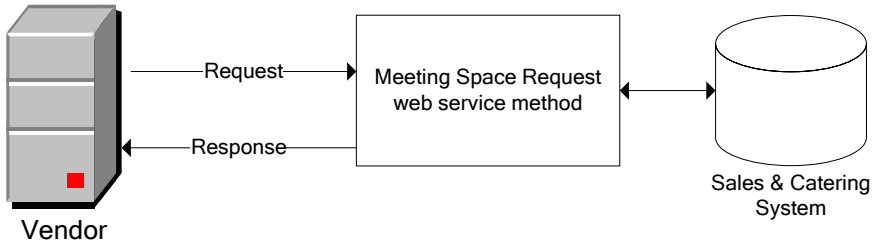
19.3 Meeting Space Request – Vendor to Sales & Catering

19.3.1 Description

This service request returns Meeting Space data for the specified property key. The caller can optionally specify the following filter criteria:

- Event key
- Whether or not the Meeting Space is an exhibit
- Whether or not the Meeting Space is postable
- A room grouping
- A date/time range

19.3.2 Service Request Flow

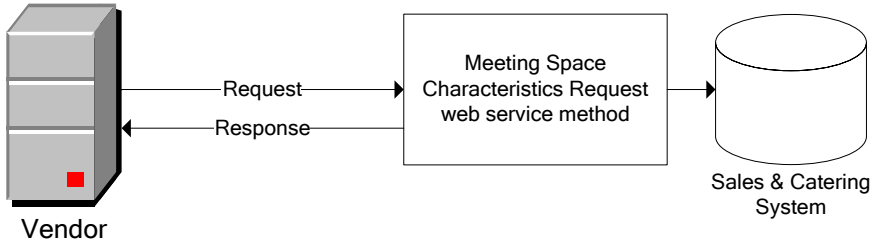


19.4 Meeting Space Characteristics Request – Vendor to Sales & Catering

19.4.1 Description

This service request return Meeting Space Characteristics for one or all properties registered with the Sales & Catering system.

19.4.2 Service Request Flow



Chapter 20 Technical Descriptions

20.1 Meeting Space Request – Vendor to Sales & Catering

20.1.1 Overview

The Vendor will post a SOAP envelope containing the XML document defined by the schema to the Sales & Catering published SOAP service.

20.1.2 Service Specifications

Service Type	SOAP
Service Name	MeetingSpaceService
Service Operation	MeetingSpaceRequest
Service Parameter Name	MeetingSpaceRequest
Response Type	xml data
Response Value	MeetingSpaceRequest_Sample.xml
Header Action:	http://htng.org/PWSWG/2007/01/DigitalSignage/MeetingSpaceRequest
URL:	nnn.nnn.nnn.nnn defined *by the IP address used to reach the S & C web service.*
Resource:	/MeetingSpaceService/Request.asmx?WSDL
Input Schema:	MeetingSpaceRequest.xsd
Output (response) Schema:	MeetingSpaceResponse.xsd, or MeetingSpaceServiceFaults.xsd when a SOAP exception is thrown.

20.1.3 Sample Request Message

MeetingSpaceRequest_Sample.xml	This file is included as Chapter 23.1.

20.1.4 Sample Response Message

MeetingSpaceResponse_Sample.xml	This file is included as Chapter 23.2.

20.1.5 Error Faults

If an error occurs, then the response xml data is not returned and a SOAP fault is thrown from the web service. The fault type can be one of the following:

Fault Type	Description
NoMeetingSpaceFoundFault	No Meeting Space entities were found for the specified input filter parameters.
InvalidDateTimeFault	An invalid date or time was specified, or the relationship between the start date and end date is not valid.
InvalidPropertyKeyFault	An invalid property key was specified.
InvalidRoomGroupingFault	An invalid room grouping was specified.
InvalidEventKeyFault	An invalid event key was specified.
DigitalSignageGenericFault	An error occurred and cannot be properly categorized by any of the previous fault types.
DateRangeTooLargeFault	The date range specified was too large to process.

20.2 Meeting Space Characteristics Request – Vendor to Sales & Catering

20.2.1 Requirements

The Vendor will post a SOAP envelope containing the XML document defined by the schema to the Sales & Catering published SOAP service.

20.2.2 Service Specifications

Service Type	SOAP
Service Name	MeetingSpaceService
Service Operation	MeetingSpaceCharacteristicsRequest
Service Parameter Name	MeetingSpaceCharacteristicsResponse
Response Type	xml data
Response Value	MeetingSpaceCharacteristicsResponse_Sample.xml
Header Action:	http://htng.org/PWSWG/2007/01/DigitalSignage/MeetingSpaceCharacteristicsRequest
URL:	nnn.nnn.nnn.nnn defined *by the IP address used to reach the S & C web service*

Resource:	/MeetingSpaceService/Request.asmx?WSDL
Input Schema:	MeetingSpaceCharacteristicsRequest.xsd
Output (response) schema:	MeetingSpaceCharacteristicsResponse.xsd, or MeetingSpaceServiceFaults.xsd when a SOAP exception is thrown.

20.2.3 Sample Request Message

MeetingSpaceCharacteristicsRequest_Sample.xml	This file is included as Chapter 23.3.

20.2.4 Sample Reponse Message

MeetingSpaceCharacteristicsResponse_Sample.xml	This file is included as Chapter 23.4.

20.2.5 Error Faults

If an error occurs, then the response xml data is not returned and a SOAP fault is thrown from the web service. The fault type can be one of the following:

Fault Type	Description
NoMeetingSpaceFoundFault	No meeting space was found in the Sales & Catering system for the specified property (if a property key was included in the request).
InvalidPropertyKeyFault	An invalid property key was specified.
DigitalSignageGenericFault	An error occurred and cannot be properly categorized by any of the previous fault types.

Chapter 21 Business Rules

21.1 Usage

1. All outbound Sales & Catering XML messages are UTF-8 encoded.
2. All inbound vendor XML messages must be UTF-8 encoded.
3. All XML should be XML encoded by the Vendor before passing that data to the Sales & Catering web service. The following are examples which should be followed for encoding the XML message. By no means does this constitute a complete list of special encoding.
    - The character < should be encoded as <.
    - The character > should be encoded as >.
    - The character & should be encoded as &.
    - All attributes should use the double quote to delimit the attribute value. If the attribute values contains a double quote, then that double quote should be encoded as ".
4. The SOAP message should follow the HTNG 2.0 draft specification. This includes the following details in the inbound request message:
    - Creation of a WS-Addressing message ID that is included in the SOAP request.
    - Provide a Sales & Catering supplied username and password token in the SOAP request using WS-Security.

21.2 Caching

The Sales & Catering system reserves the right to make a configurable response cache to prevent excessive requests from Vendor systems.

Chapter 22 MeetingSpaceService WSDL

The wsdl definition for this service is listed below:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- edited with XMLSPY v2004 rel. 4 U (http://www.xmlspy.com) by Kristen Acheson (Newmarket International, Inc.) -->
<!-- This wsdl defines the HTNG 2.0 Digital Signage web service.-->
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://htng.org/PWSWG/2007/01/DigitalSignage" xmlns:s1="http://htng.org/PWSWG/2007/01/DigitalSignage/MeetingSpaceRequest/Types"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:s2="http://htng.org/PWSWG/2007/01/DigitalSignage/MeetingSpaceResponse/Types"
xmlns:s3="http://htng.org/PWSWG/2007/01/DigitalSignage/MeetingSpaceCharacteristicsRequest/Types"
xmlns:s4="http://htng.org/PWSWG/2007/01/DigitalSignage/MeetingSpaceCharacteristicsResponse/Types"
xmlns:s5="http://htng.org/PWSWG/2007/01/DigitalSignage/Faults/Types" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://htng.org/PWSWG/2007/01/DigitalSignage">
        <wsdl:types>
                <s:schema>
                        <s:import namespace="http://htng.org/PWSWG/2007/01/DigitalSignage/MeetingSpaceCharacteristicsRequest/Types"
schemaLocation="..\Schemas\MeetingSpaceCharacteristicsRequest.xsd"/>
                        <s:import namespace="http://htng.org/PWSWG/2007/01/DigitalSignage/MeetingSpaceCharacteristicsResponse/Types"
schemaLocation="..\Schemas\MeetingSpaceCharacteristicsResponse.xsd"/>
                        <s:import namespace="http://htng.org/PWSWG/2007/01/DigitalSignage/MeetingSpaceRequest/Types"
schemaLocation="..\Schemas\MeetingSpaceRequest.xsd"/>
                        <s:import namespace="http://htng.org/PWSWG/2007/01/DigitalSignage/MeetingSpaceResponse/Types"
schemaLocation="..\Schemas\MeetingSpaceResponse.xsd"/>
                        <s:import namespace="http://htng.org/PWSWG/2007/01/DigitalSignage/Faults/Types"
schemaLocation="..\Schemas\MeetingSpaceServiceFaults.xsd"/>
                </s:schema>
        </wsdl:types>
        <wsdl:message name="MeetingSpaceSoapIn">
                <wsdl:part name="MeetingSpaceRequest" element="s1:MeetingSpaceRequest"/>
        </wsdl:message>
        <wsdl:message name="MeetingSpaceSoapOut">
                <wsdl:part name="MeetingSpaceResponse" element="s2:MeetingSpaceResponse"/>
        </wsdl:message>
        <wsdl:message name="MeetingSpaceCharacteristicsSoapIn">
                <wsdl:part name="MeetingSpaceCharacteristicsRequest" element="s3:MeetingSpaceCharacteristicsRequest"/>
        </wsdl:message>
        <wsdl:message name="MeetingSpaceCharacteristicsSoapOut">
                <wsdl:part name="MeetingSpaceCharacteristicsResponse" element="s4:MeetingSpaceCharacteristicsResponse"/>
        </wsdl:message>
        <wsdl:message name="DigitalSignageGenericFault">
                <wsdl:part name="parameters" element="s5:DigitalSignageGenericFault"/>
        </wsdl:message>
        <wsdl:message name="NoMeetingSpaceFoundFault">
                <wsdl:part name="parameters" element="s5:NoMeetingSpaceFoundFault"/>
```

```
</wsdl:message>
<wsdl:message name="InvalidDateTimeFault">
        <wsdl:part name="parameters" element="s5:InvalidDateTimeFault"/>
</wsdl:message>
<wsdl:message name="DateRangeTooLargeFault">
        <wsdl:part name="parameters" element="s5:DateRangeTooLargeFault"/>
</wsdl:message>
<wsdl:message name="InvalidPropertyKeyFault">
        <wsdl:part name="parameters" element="s5:InvalidPropertyKeyFault"/>
</wsdl:message>
<wsdl:message name="InvalidRoomGroupingFault">
        <wsdl:part name="parameters" element="s5:InvalidRoomGroupingFault"/>
</wsdl:message>
<wsdl:message name="InvalidEventKeyFault">
        <wsdl:part name="parameters" element="s5:InvalidEventKeyFault"/>
</wsdl:message>
<wsdl:message name="NoMeetingRoomsFoundFault">
        <wsdl:part name="parameter" element="s5:NoMeetingRoomsFoundFault"/>
</wsdl:message>
<wsdl:portType name="MeetingSpaceServiceSoap">
        <wsdl:operation name="MeetingSpaceRequest">
                <wsdl:input message="tns:MeetingSpaceSoapIn"/>
                <wsdl:output message="tns:MeetingSpaceSoapOut"/>
                <wsdl:fault name="DigitalSignageGenericFault" message="tns:DigitalSignageGenericFault"/>
                <wsdl:fault name="NoMeetingSpaceFoundFault" message="tns:NoMeetingSpaceFoundFault"/>
                <wsdl:fault name="InvalidDateTimeFault" message="tns:InvalidDateTimeFault"/>
                <wsdl:fault name="DateRangeTooLargeFault" message="tns:DateRangeTooLargeFault"/>
                <wsdl:fault name="InvalidPropertyKeyFault" message="tns:InvalidPropertyKeyFault"/>
                <wsdl:fault name="InvalidRoomGroupingFault" message="tns:InvalidRoomGroupingFault"/>
                <wsdl:fault name="InvalidEventKeyFault" message="tns:InvalidEventKeyFault"/>
        </wsdl:operation>
        <wsdl:operation name="MeetingSpaceCharacteristicsRequest">
                <wsdl:input message="tns:MeetingSpaceCharacteristicsSoapIn"/>
                <wsdl:output message="tns:MeetingSpaceCharacteristicsSoapOut"/>
                <wsdl:fault name="DigitalSignageGenericFault" message="tns:DigitalSignageGenericFault"/>
                <wsdl:fault name="InvalidPropertyKeyFault" message="tns:InvalidPropertyKeyFault"/>
                <wsdl:fault name="NoMeetingRoomsFoundFault" message="tns:NoMeetingRoomsFoundFault"/>
        </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="MeetingSpaceServiceSoap" type="tns:MeetingSpaceServiceSoap">
        <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="MeetingSpaceRequest">
                <soap:operation soapAction="http://htng.org/PWSWG/2007/01/DigitalSignage#MeetingSpaceRequest" style="document"/>
                <wsdl:input>
                        <soap:body use="literal"/>
                </wsdl:input>
                <wsdl:output>
```

```
                        <soap:body use="literal"/>
                    </wsdl:output>
                    <wsdl:fault name="DigitalSignageGenericFault">
                        <wsdl:documentation>Contains an error code and message that explains the error details.  It is used when no other defined faults
are relevant.</wsdl:documentation>
                    </wsdl:fault>
                    <wsdl:fault name="NoMeetingSpaceFoundFault">
                        <wsdl:documentation>No meeting space items were found for the specified filter criteria.</wsdl:documentation>
                    </wsdl:fault>
                    <wsdl:fault name="InvalidDateTimeFault">
                        <wsdl:documentation>Invalid date or time information was specified.</wsdl:documentation>
                    </wsdl:fault>
                    <wsdl:fault name="DateRangeTooLargeFault">
                        <wsdl:documentation>The date range specified was too large to process.</wsdl:documentation>
                    </wsdl:fault>
                    <wsdl:fault name="InvalidPropertyKeyFault">
                        <wsdl:documentation>Invalid property key specified.</wsdl:documentation>
                    </wsdl:fault>
                    <wsdl:fault name="InvalidRoomGroupingFault">
                        <wsdl:documentation>Invalid room grouping name specified, or the Sales and Catering system does not support room
groupings.</wsdl:documentation>
                    </wsdl:fault>
                    <wsdl:fault name="InvalidEventKeyFault">
                        <wsdl:documentation>Invalid event key specified.</wsdl:documentation>
                    </wsdl:fault>
                </wsdl:operation>
                <wsdl:operation name="MeetingSpaceCharacteristicsRequest">
                    <soap:operation soapAction="http://htng.org/PWSWG/2007/01/DigitalSignage#MeetingSpaceCharacteristicsRequest" style="document"/>
                    <wsdl:input>
                        <soap:body use="literal"/>
                    </wsdl:input>
                    <wsdl:output>
                        <soap:body use="literal"/>
                    </wsdl:output>
                    <wsdl:fault name="DigitalSignageGenericFault">
                        <wsdl:documentation>Contains an error code and message that explains the error details.  It is used when no other defined faults
are relevant.</wsdl:documentation>
                    </wsdl:fault>
                    <wsdl:fault name="NoMeetingRoomsFoundFault">
                        <wsdl:documentation>No meeting rooms were found.</wsdl:documentation>
                    </wsdl:fault>
                    <wsdl:fault name="InvalidPropertyKeyFault">
                        <wsdl:documentation>Invalid property key specified.</wsdl:documentation>
                    </wsdl:fault>
                </wsdl:operation>
            </wsdl:binding>
            <wsdl:service name="MeetingSpaceService">
```

```
            <wsdl:port name="MeetingSpaceServiceSoap" binding="tns:MeetingSpaceServiceSoap">
                <soap:address location="http://localhost/MeetingSpaceService/Request.asmx"/>
            </wsdl:port>
        </wsdl:service>
</wsdl:definitions>
```

Chapter 23 SOAP Examples

This section provides sample SOAP transactions that are used to communicate with the MeetingSpaceService web service.

23.1 Meeting Space Request

Below is an example of a raw SOAP request for Meeting Space data.

Sample File: **MeetingSpaceRequest_Sample.xml**


```
    <?xml version="1.0" encoding="utf-8" ?>
- <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
    open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
- <soap:Header>
  <wsa:Action>http://htng.org/PWSWG/2007/01/DigitalSignage#MeetingSpaceRequest</wsa:Action>
  <wsa:MessageID>urn:uuid:44088898-6926-4537-b1a3-4acc8e9f71d0</wsa:MessageID>
- <wsa:ReplyTo>
  <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
    </wsa:ReplyTo>
  <wsa:To>http://localhost:2025/MeetingSpaceProviderService/Request.asmx</wsa:To>
- <wsse:Security soap:mustUnderstand="1">
- <wsu:Timestamp wsu:Id="Timestamp-e157d58f-6589-4b3e-b2a9-5510dd9f270f">
  <wsu:Created>2008-04-17T15:35:25Z</wsu:Created>
  <wsu:Expires>2008-04-17T15:40:25Z</wsu:Expires>
    </wsu:Timestamp>
- <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
    843aeb4f-07de-4c78-a438-e051d3baa764">
  <wsse:Username>TEST</wsse:Username>
  <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
    1.0#PasswordText">TEST_PASS</wsse:Password>
  <wsse:Nonce>9fY7BamB3GO47y+BsO8jig==</wsse:Nonce>
  <wsu:Created>2008-04-17T15:35:25Z</wsu:Created>
    </wsse:UsernameToken>
    </wsse:Security>
    </soap:Header>
- <soap:Body>
- <MeetingSpaceRequest propertyKey="3917" eventKey="" xmlns="http://htng.org/PWSWG/2007/01/DigitalSignage/MeetingSpaceRequest/Types">
  <DateRange startDateTime="2008-04-17T08:00:00" endDateTime="2008-04-17T23:00:00" />
    </MeetingSpaceRequest>
    </soap:Body>
    </soap:Envelope>
```

23.2 Meeting Space Response

This example shows the Meeting Space response SOAP message. Note that the wsa:RelatesTo guid in the SOAP header will be the same value that was sent up as the wsa:MessageID in the request.

NOTE: The *isResponseComplete* attribute at the top level MeetingSpaceResponse node is used to tell the caller if all of the requested data was returned. There can be cases where all of the data is not returned because the calller did not use a specific enough set of input filter parameters.

Sample File: **MeetingSpaceResponse_Sample.xml**


```
  <?xml version="1.0" ?>
- <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
    open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
- <soap:Header>
  <wsa:Action>http://htng.org/PWSWG/2007/01/DigitalSignage#MeetingSpaceRequestResponse</wsa:Action>
  <wsa:MessageID>urn:uuid:107a8d1a-51af-4f06-a6ef-dc9cc50a88fc</wsa:MessageID>
  <wsa:RelatesTo>urn:uuid:44088898-6926-4537-b1a3-4acc8e9f71d0</wsa:RelatesTo>
  <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
- <wsse:Security>
- <wsu:Timestamp wsu:Id="Timestamp-69a205cf-5348-48be-bf4d-edde0f2dd758">
  <wsu:Created>2008-04-17T15:35:25Z</wsu:Created>
  <wsu:Expires>2008-04-17T15:40:25Z</wsu:Expires>
    </wsu:Timestamp>
    </wsse:Security>
    </soap:Header>
- <soap:Body>
- <MeetingSpaceResponse xmlns="http://htng.org/PWSWG/2007/01/DigitalSignage/MeetingSpaceResponse/Types">
- <Property propertyKey="3917" propertyName="Newmarket Hotel & Spa">
- <Event eventKey="44.42" eventName="Quarterly Meeting" eventLogo="" eventPostAs="Quarterly Meeting">
  <Group name="Newmarket Travel Agency" key="4567.42" />
- <MeetingSpace startDateTime="2008-04-17T11:21:20" endDateTime="2008-04-17T23:00:00" roomKey="123.42" roomName="BallRoom 2"
      meetingKey="123.42.222.42" meetingName="Dinner" meetingPostAs="Dinner Reception" isPostable="true" isExhibit="false">
  <BackupMeetingRoom roomName="Backup Meeting Room Name" roomKey="1111.42" />
  <OverflowMeetingRoom roomName="West Wing Corridor" roomKey="1112" />
    </MeetingSpace>
  <MeetingSpace startDateTime="0001-01-01T00:00:00" endDateTime="2008-04-17T17:00:00" roomKey="124.42" roomName="BallRoom 1"
      meetingKey="124.42.222.42" meetingName="Quarterly Meeting" meetingPostAs="Quarterly Meeting" isPostable="true" isExhibit="false" />
    </Event>
- <Event eventKey="4224.42" eventName="Kickoff Meeting" eventLogo="" eventPostAs="Kickoff Meeting">
  <Group name="Portsmouth Travel Agency" key="6789.42" />
- <MeetingSpace startDateTime="2008-05-17T11:21:20" endDateTime="2008-05-17T23:00:00" roomKey="123.42" roomName="BallRoom 2"
      meetingKey="123.42.222.42" meetingName="Dinner" meetingPostAs="Dinner Reception" isPostable="true" isExhibit="false">
  <BackupMeetingRoom roomName="Backup Meeting Room Name" roomKey="1111.42" />
```

```
<OverflowMeetingRoom roomName="West Wing Corridor" roomKey="1112" />
  </MeetingSpace>
<MeetingSpace startDateTime="0001-01-01T00:00:00" endDateTime="2008-04-17T17:00:00" roomKey="124.42" roomName="BallRoom 1"
   meetingKey="124.42.222.42" meetingName="Quarterly Meeting" meetingPostAs="Quarterly Meeting" isPostable="true" isExhibit="false" />
  </Event>
  </Property>
  </MeetingSpaceResponse>
  </soap:Body>
  </soap:Envelope>
```


23.3 Meeting Space Characteristics Request

This example shows the Meeting Space Characteristics request SOAP message.

Sample File: **MeetingSpaceCharacteristicsRequest_Sample.xml**


```
    <?xml version="1.0" encoding="utf-8" ?>
- <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
    open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
- <soap:Header>
  <wsa:Action>http://htng.org/PWSWG/2007/01/DigitalSignage#MeetingSpaceCharacteristicsRequest</wsa:Action>
  <wsa:MessageID>urn:uuid:2902d88c-58c1-44de-88ba-1c3182fac2ab</wsa:MessageID>
- <wsa:ReplyTo>
  <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
    </wsa:ReplyTo>
  <wsa:To>http://localhost:2025/MeetingSpaceProviderService/Request.asmx</wsa:To>
- <wsse:Security soap:mustUnderstand="1">
- <wsu:Timestamp wsu:Id="Timestamp-31276e8c-193e-49bc-8ddc-fe146e645506">
  <wsu:Created>2008-04-17T17:25:42Z</wsu:Created>
  <wsu:Expires>2008-04-17T17:30:42Z</wsu:Expires>
    </wsu:Timestamp>
- <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="SecurityToken-
    0dae464b-4c19-4fb3-90c3-f366762e76c6">
  <wsse:Username>TEST</wsse:Username>
  <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
    1.0#PasswordText">TEST_PASS</wsse:Password>
  <wsse:Nonce>G01cMzwEIcj/lFoWtn8BJA==</wsse:Nonce>
  <wsu:Created>2008-04-17T17:25:42Z</wsu:Created>
    </wsse:UsernameToken>
    </wsse:Security>
    </soap:Header>
```

```
- <soap:Body>
  <MeetingSpaceCharacteristicsRequest propertyKey="3917"
      xmlns="http://htng.org/PWSWG/2007/01/DigitalSignage/MeetingSpaceCharacteristicsRequest/Types" />
    </soap:Body>
    </soap:Envelope>
```


23.4 Meeting Space Characteristics Response

This is an example of a raw SOAP response messsage for Meeting Space Characteristics data.

Sample File: **MeetingSpaceCharacteristicsResponse_Sample.xml**


```
    <?xml version="1.0" encoding="utf-8" ?>
- <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
      xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
      open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
- <soap:Header>
  <wsa:Action>http://htng.org/PWSWG/2007/01/DigitalSignage#MeetingSpaceCharacteristicsRequestResponse</wsa:Action>
  <wsa:MessageID>urn:uuid:90ee13e9-ba65-423b-9596-9f3e30b145af</wsa:MessageID>
  <wsa:RelatesTo>urn:uuid:2902d88c-58c1-44de-88ba-1c3182fac2ab</wsa:RelatesTo>
  <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
- <wsse:Security>
- <wsu:Timestamp wsu:Id="Timestamp-d26c2884-0df1-4e80-954b-d081bf205bda">
  <wsu:Created>2008-04-17T17:25:44Z</wsu:Created>
  <wsu:Expires>2008-04-17T17:30:44Z</wsu:Expires>
    </wsu:Timestamp>
    </wsse:Security>
    </soap:Header>
- <soap:Body>
- <MeetingSpaceCharacteristicsResponse isResponseComplete="true"
      xmlns="http://htng.org/PWSWG/2007/01/DigitalSignage/MeetingSpaceCharacteristicsResponse/Types">
- <Property propertyKey="3917" propertyName="Newmarket Hotel & Conference Center">
  <MeetingSpaceCharacteristics roomKey="123.42" roomName="BallRoom 2" floorNumber="2" floorDescription="2nd Floor" roomGrouping="North Hall"
      directions="Proceed down the north hall and take a right." isDivisible="true" parentRoomKey="" mapReferenceNum="225" />
  <MeetingSpaceCharacteristics roomKey="223.42" roomName="BallRoom 1" floorNumber="1" floorDescription="Main Floor" roomGrouping="North Hall"
      directions="Proceed down the north hall and take a right." isDivisible="true" parentRoomKey="" mapReferenceNum="125" />
  <MeetingSpaceCharacteristics roomKey="100.42" roomName="Winchester Meeting Room" floorNumber="1" floorDescription="Main Floor"
      roomGrouping="West Wing" directions="Proceed down the main hall and take a left to west wing." parentRoomKey="" mapReferenceNum="110" />
  <MeetingSpaceCharacteristics roomKey="100.42" roomName="Meeting Room A" floorNumber="3" floorDescription="3rd Floor" roomGrouping="Meeting
      Rooms" directions="Off the 3rd Floor elevator to the left." isDivisible="true" parentRoomKey="" mapReferenceNum="310" />
    </Property>
    </MeetingSpaceCharacteristicsResponse>
    </soap:Body>
```

```
</soap:Envelope>
```

23.5 Meeting Space Service SOAP Faults Returned to Client

When an error occurs within the MeetingSpaceService web service, it throws a custom exception back to the caller. The possible types of exceptions returned are defined as *Error Faults* in Sections 18.1 and 18.2. The definitions for these faults are defined in the MeetingSpaceServiceFaults.xsd schema, which is referenced from the MeetingSpaceService.wsdl.

This is an example of a SOAP envelope for Meeting Space response data where an exception has been thrown on the server because the end date ocurred after the start date.

Sample File: **MeetingSpaceResponse_SOAP_Fault_Sample.xml**


```
<?xml version="1.0" encoding="UTF-8" ?>
- <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
    open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
- <soap:Header>
  <wsa:Action>http://schemas.xmlsoap.org/ws/2004/08/addressing/fault</wsa:Action>
  <wsa:MessageID>urn:uuid:3979f487-b545-4926-9c16-cbacfbf13795</wsa:MessageID>
  <wsa:RelatesTo>urn:uuid:342f7af8-d38a-47ab-b83e-ea63bd90473c</wsa:RelatesTo>
  <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
- <wsse:Security>
- <wsu:Timestamp wsu:Id="Timestamp-89d0bd3d-1512-41f1-a352-43e5af5d9117">
  <wsu:Created>2007-01-19T14:31:20Z</wsu:Created>
  <wsu:Expires>2007-01-19T14:36:20Z</wsu:Expires>
    </wsu:Timestamp>
    </wsse:Security>
    </soap:Header>
- <soap:Body>
- <soap:Fault>
  <faultcode>soap:Client</faultcode>
  <faultstring>WebServices.Faults.InvalidDateTimeFaultException: An exception occurred.</faultstring>
- <detail>
- <InvalidDateTimeFault xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns="http://htng.org/PWSWG/2007/01/DigitalSignage/Faults/Types">
  <Details>The end date must be greater than or equal to the start date.</Details>
    </InvalidDateTimeFault>
    </detail>
    </soap:Fault>
    </soap:Body>
    </soap:Envelope>
```

Part 4 ***Open Data Exchange Version 1.0***

Chapter 24 Document History

24.1 Document Changes

Version	Date	Author	Comments
1.0.1	03 December 2007	Paul Armstrong	Initial document available for discussion, To be done - provide: 1. Provide element and attribute descriptions include status code definitions. 2. Provide samples 3. Provide comments
1.0.2	06 December 2007	Paul Armstrong	Added sections required by standard format. Added element and attribute descriptions.
1.0.3	13 December 2007	Workgroup	Changed message names to include HTNG_ODX prefix. Added text to Sec. 5. Changed URIs to reflect htng.org/PWS/2008A designators.
1.1.0	13 January 2008	Brian Alessi	1. Updated FileTransfer.xsd to include file of type base64binary. 2. Created use cases for PullFile and PushFile messages. 3. Inserted XML descriptions of data with PullFileRequest, PullFileResponse, PushFileRequest, and PushFileResponse. 4. Inserted Request and Response SOAP examples from in-house implementation for both PullFile and PushFile.

Chapter 25 Document Information

25.1 Document Purpose

The document defines the Hotel Technology Next Generation ("HTNG") Open Data Exchange Web Services messages.

25.2 Scope

The document defines the web services requests and responses that must be implemented to support ODX. The document also includes definitions, use cases and messages flows associated with ODX data transfers.

25.3 Audience

This document is intended for use by developers and systems integrators who wish to use ODX to implement a standard data transfer mechanism. Other IT professionals will find this document useful in gaining a general understanding of HTNG ODX.

25.4 Overview

ODX messages support "push" and "pull" data transfers. An application may "push" or send data file(s) unsolicited, to a "destination" application. Alternatively an application may "pull" or request data file(s) from a "source" application.

The message formats may be extended and designed to support many diverse use cases.

The WSDL for the HTNG Open Data Exchange defines the "ODXService" web service. This web service consists of two functions defined in one port type. The port type is:

- FileTransferPort
  - Generic file transfer functions which may be implemented by any two applications wishing to transfer one or more files of any format.

25.4.1 Supporting Schemas

The Open Data Exchange Web Service imports 1 schema which defines data payloads required by the functional messages. These are:

- FileTransfer.xsd
  - Defines request and response message contents and options.

As part of the standard HTNG specification it is also required to import 1 custom header schema as follows:

- HTNG_AsyncHeaders.xsd

- Defines the standard header elements required for complying with HTNG specification.

25.5 Document Terms

For the purpose of this document the following terms have been defined as follows:

Term	Definition
HTNG	Hotel Technology Next Generation
MTOM	Message Transmission Optimization Mechanism – Used by ODX to optimize file transfers when using web services. MTOM optimizes the transfer of base 64 encoded data (xs:base64binary) by automatically including it as a binary attachment to the message.
ODX	Open Data Exchange
WSDL	Web Services Description Language
XML	Extensible Markup Language – This is a general-purpose markup language for creating special-purpose markup languages, capable of describing different kinds of data.
XSD	XML Schema Definition – A language for specifying the grammar of the markup allowed in an XML file. It describes the structure of an XML document

25.6 Referenced Documents

Messages defined in this specification will be implemented using the HTNG Web Service Framework V2.0.

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

Name	Location
HTNG Property Web Services Framework 2.0	http://www.htng.org

Chapter 26 Business Process

26.1 Overview

Today, much inter-application communication is achieved by transferring data files of various formats using many transport mechanisms. The transport mechanisms and processes vary by application/vendor pair and in practice they are often primitive and usually very difficult to monitor and manage. For hoteliers, this impacts property system reliability, increases support costs and makes technology evolution a costly headache.

There is a need to establish a simple, consistent and manageable inter-application file transfer mechanism. This mechanism needs to be platform independent and compatible with tomorrow's Enterprise Service Bus architectures.

26.2 Roles

There are two roles; a File Transfer Requestor (ODX Service Requestor) and a File Transfer Responder (ODX Service Provider).

26.3 Behavior

The File Transfer Requestor is responsible for requesting a file transfer, by requesting specific files or providing specific files.
The File Transfer Responder either receives the files included in the request or responds with message that includes the requested files. The File Transfer Responder is the Web Service provider.

26.4 Use Cases

26.4.1 PullFileRequest / PullFileResponse

The PullFileRequest will give consumers the ability to obtain one or more files from a source system. These files can be large in nature as the following schemas support binary transfer of the files from the source system. This is so the entire response doesn't need to be stored in memory. A consumer can initiate a PullFileRequest to request one or more files from a providing system. Based on the implementation, the location of the files requested should be known by the consumer or should be implied when making a call for a file. The PullFileResponse will contain all file information as defined in FileTransfer.xsd.

- ODX Service Requestor makes a PullFileRequest to the ODX Service Provider.
- The ODX Service Provider reviews the request and collects all file requested.
- The files are converted to byte arrays so they may be transfer back to the ODX Service Requestor.
- The ODX Service Provider delivers the PullFileResponse object back to the ODX Service Requestor.
- The ODX Service Requestor reviews the response for the data which was requested and any errors.

26.4.2 PushFileRequest/PushFileResponse

The PushFileRequest will allow consumers to package multiple files into one message for transfer to a destination (in this case a providing) system. During the data transfer there may exist error messages in the PushFileResponse which can be mimicked back to the consumer.

- The ODX Service Requestor packages a set of file on their server.
- The ODX Service Requestor converts the file to byte arrays for transfer to the ODX Service Provider.
- Using MTOM the ODX Service Requsestor consumes the PushFileRequest method from the ODX Service Provider.
- The ODX Service Provider accepts the incoming PushFileRequest and forms a PushFileResponse object.
- The ODX Service Provider delivers the PushFileResponse object back to the ODX Service Requestor.
- The ODX Service Requestor reviews the response for errors.

In all methods of communication the File contents are a byte array transferred using MTOM for large file transfer support. These byte arrays can either be exported to represent the documents as they were in the source system or they can be consumed within the destination service/application. This would have to be an agreement made between vendor systems outside of this specification. There is not contention for how the data is manipulated after it has reached its destination. It is intended the destination service/application mine the data as needed.

Chapter 27 Schemas

27.1 Open Data Exchange

27.1.1 Soap Header

All messages must include a soap header that conforms to the WSAddressing and WSSecurity specifications. Receiving systems may optionally require that the wsa:To element match a specific destination address, and therefore, this element should be configurable at run time. In addition, all sending systems must identify themselves by specifying a wsa:From element unique for their system. Typically, this is in the form of "URN:*<system>*". The wsa:ReplyTo address may be the anonymous form described in the August, 2004 specification.

WSSecurity may be optionally implemented. Providers are only required to support basic plain text authentication with a username and password.

The user credentials will be limited to a single login per system, agreed between two vendors. There is no requirement to support multiple user logins from a single vendor through this interface.

27.1.1.1 Sample Header Message

```
<soap:Header>
<wsa:Action>http://htng.org/PWS/2008A/OpenDataExchange#HTNG_ODXPushFile</wsa:Action>
<wsa:From>
<wsa:Address>urn:SourceApplication</wsa:Address>
</wsa:From>
<wsa:MessageID>urn:uuid:e63d962e-94b6-434c-89ea-1c5ae2e0f8ba</wsa:MessageID>
<wsa:ReplyTo>
<wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
</wsa:ReplyTo>
<wsa:To>http://www.othersystem.com/HTNGFileTransfer/</wsa:To>
<wsse:Security soap:mustUnderstand="1">
<wsu:Timestamp wsu:Id="Timestamp-015e2941-114a-466e-87b6-8c431b9f5c44">
<wsu:Created>2006-10-26T12:49:17Z</wsu:Created>
<wsu:Expires>2006-10-26T12:54:17Z</wsu:Expires>
</wsu:Timestamp>
<wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="SecurityToken-627f1ab1-338a-
451f-9829-84f248e57ad8">
<wsse:Username>HTNG</wsse:Username>
<wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">HTNG</wsse:Password>
<wsse:Nonce>qls5nr9rM7VaUYAwhrHSoA==</wsse:Nonce>
<wsu:Created>2006-10-26T12:49:17Z</wsu:Created>
</wsse:UsernameToken>
</wsse:Security>
</soap:Header>
```

27.1.2 File Transfer Port

The File Transfer Port defines two functions which should be implemented by File Source and File Destination Systems.

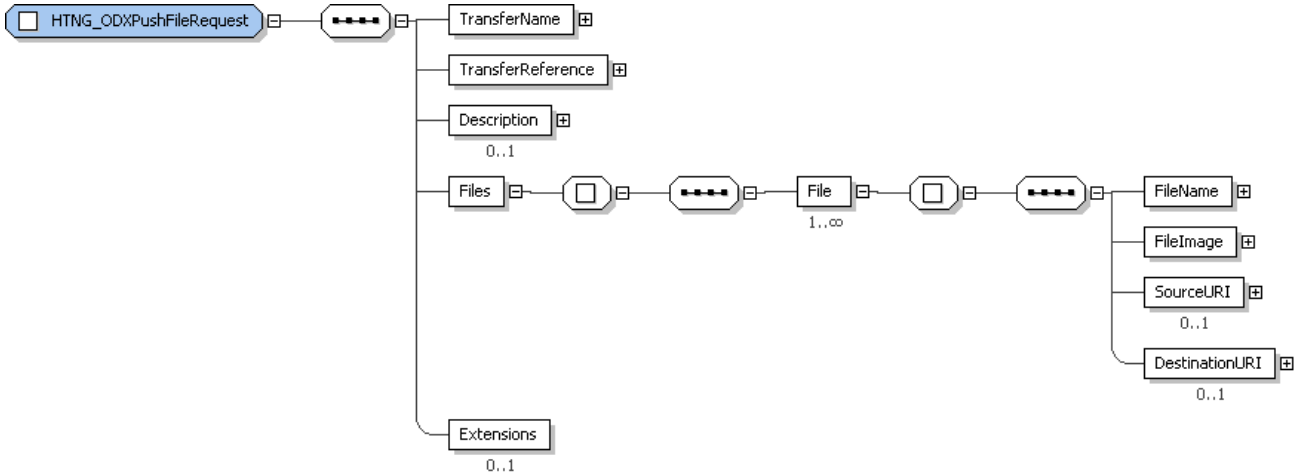
PushFile
Function for a source application to request file(s) be transferred to a destination application. The destination application is the service provider,

PullFile
Function for a destination application to request file(s) from a source application. The source application is the service provider.

27.1.3 PushFile

A Push File request message is submitted by a source application to a destination application in order to transfer one or more files from the source application to the destination application. MTOM (Message Transmission Optimization Mechanism) should be enabled.

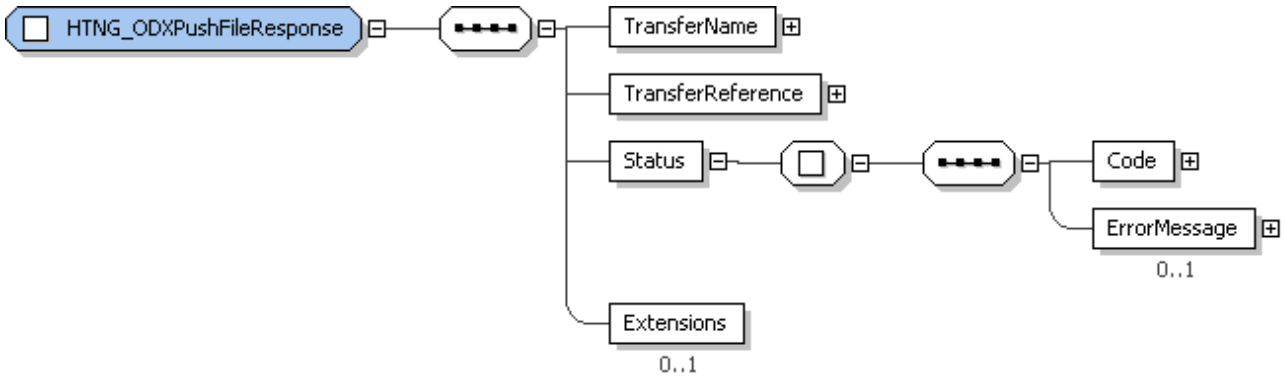
Port	FileTransferPort
Operation	PushFile
Soap Action	http://htng.org/PWS/2008A/OpenDataExchange#PushFile
Input	HTNG_ODXPushFileRequest
Output	HTNG_ODXPushFileResponse
Primary Schema	FileTransfer.xsd
Role(s) Implemented	File Transfer Requestor, Fie Transfer Responder



Element | @Attribute	Num	Description/Contents
HTNG_ODXPushFileRequest	1	Root element of the message
TransferName	1	Arbitrary name given to this transfer, should be meaningful to the responder.
TransferReference	1	Specific transfer reference, should be unique for each request message.
Description	0..1	Optional description.
Files	1	Element contains the files being transferred.
File	1..N	Contains a file being transferred.
FileName	1	The filename of the file being transferred.
FileImage	1	The file's contents
SourceURI	0..1	Optional element containing the URI from where the file was sourced by the requestor.
DestinationURI	0..1	Optional element containing a URI that defines where the responder should place the file.
Extensions	0..1	Optional element into which Requestor and Responder may place mutually agreed extensions.

PushFileRequest

```
<PushFile xmlns="http://htng.org/PWS/2008A/ODXService#PushFile">
      <Request>
        <TransferName>string</TransferName>
        <TransferReference>string</TransferReference>
        <Description>string</Description>
        <Files>
          <File>
            <FileName>string</FileName>
            <FileImage>base64Binary</FileImage>
            <SourceURI>string</SourceURI>
            <DestinationURI>string</DestinationURI>
            <File>base64Binary</File>
          </File>
          <File>
            <FileName>string</FileName>
            <FileImage>base64Binary</FileImage>
            <SourceURI>string</SourceURI>
            <DestinationURI>string</DestinationURI>
            <File>base64Binary</File>
          </File>
        </Files>
        <Extensions />
      </Request>
    </PushFile>
```



Element | @Attribute	Num	Description/Contents
HTNG_ODXPushFileResponse	1	Root element of the message
TransferName	1	Arbitrary name given to this transfer, should contain the same value as the request message.
TransferReference	1	Specific transfer reference, should contain the same value as the request message.
Status	1	Contains status information.
Code	1	Contains: 0 = Successful < 0: Error
ErrorMessage	0..1	If Code = Error, this element contains a message describing the error reason.
Extensions	0..1	Optional element into which Requestor and Responder may place mutually agreed extensions.

PushFileResponse

```
<PushFileResponse xmlns="http://htng.org/PWS/2008A/ODXService#PushFile">
      <PushFileResult>
        <TransferName>string</TransferName>
        <TransferReference>string</TransferReference>
        <Status>
          <Code>string</Code>
          <ErrorMessage>string</ErrorMessage>
        </Status>
        <Extensions />
      </PushFileResult>
    </PushFileResponse>
```

Sample Message

[*Request*]

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <soap:Header>
          <myHeader>
            <CorrelationID xmlns="http://htng.org/PWSWG/2007/02/AsyncHeaders">urn:uuid:1e1db2be-32eb-43fc-a34f-
c21191672c8a</CorrelationID>
            <ReplyTo xmlns="http://htng.org/PWSWG/2007/02/AsyncHeaders">
              <Address xmlns="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://localhost/ConsumerService/Service.asmx</Address>
            </ReplyTo>
            <FaultTo xmlns="http://htng.org/PWSWG/2007/02/AsyncHeaders">
              <Address xmlns="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://localhost/ConsumerService/Service.asmx</Address>
            </FaultTo>
          </myHeader>
          <wsa:Action>http://htng.org/PWS/2008A/ODXService#PullFile/PullFile</wsa:Action>
          <wsa:MessageID>urn:uuid:fa161e28-bc73-4f92-8c47-815fdfc42677</wsa:MessageID>
          <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
          </wsa:ReplyTo>
          <wsa:To>http://localhost:4060/ODXProviderService/ODXServices.asmx</wsa:To>
          <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-7cf82de0-0fb6-48a2-bc83-36ed50fbea21">
              <wsu:Created>2008-01-14T02:50:04Z</wsu:Created>
              <wsu:Expires>2008-01-14T02:55:04Z</wsu:Expires>
            </wsu:Timestamp>
          </wsse:Security>
        </soap:Header>
        <soap:Body>
          <PullFile xmlns="http://htng.org/PWS/2008A/ODXService#PullFile">
            <Request>
              <TransferName>Test Transfer</TransferName>
              <TransferReference>http://htng.org/PWS/2008A/ODXService</TransferReference>
              <Description>Simple pull file test</Description>
              <Files>
                <File>
                  <FileName>test.txt</FileName>
                  <SourceURI>http://htng.org/PWS/2008A/ODXService/Consumer</SourceURI>
                  <DestinationURI>http://htng.org/PWS/2008A/ODXService/Producer</DestinationURI>
                </File>
              </Files>
            </Request>
          </PullFile>
        </soap:Body>
      </soap:Envelope>
```

[*Response*]

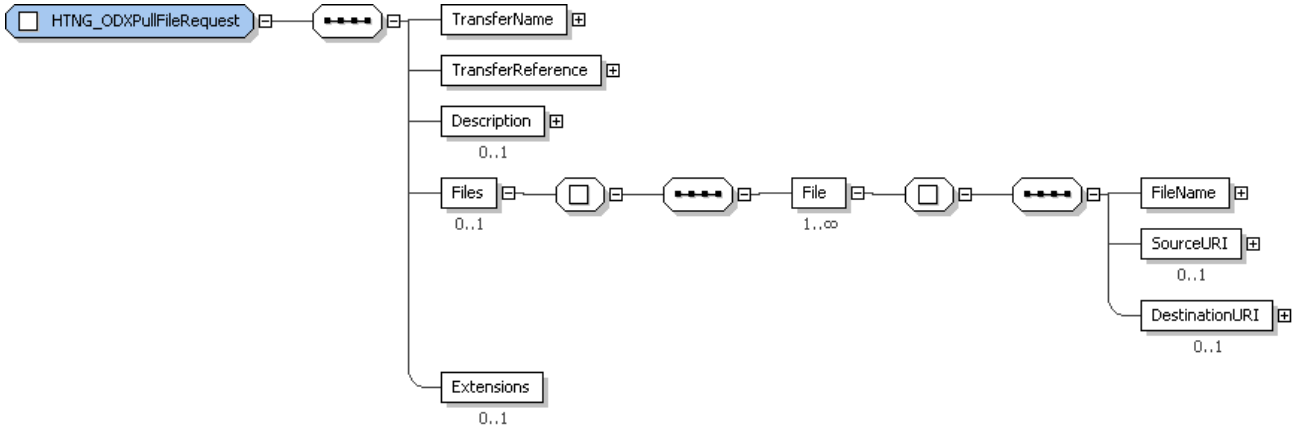

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <soap:Header>
          <myHeader>
            <CorrelationID xmlns="http://htng.org/PWSWG/2007/02/AsyncHeaders">urn:uuid:1e1db2be-32eb-43fc-a34f-
c21191672c8a</CorrelationID>
            <ReplyTo xmlns="http://htng.org/PWSWG/2007/02/AsyncHeaders">
              <Address xmlns="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://localhost/ConsumerService/Service.asmx</Address>
            </ReplyTo>
            <FaultTo xmlns="http://htng.org/PWSWG/2007/02/AsyncHeaders">
              <Address xmlns="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://localhost/ConsumerService/Service.asmx</Address>
            </FaultTo>
          </myHeader>
          <wsa:Action>http://htng.org/PWS/2008A/ODXService#PullFile/PullFileResponse</wsa:Action>
          <wsa:MessageID>urn:uuid:fee0e8a9-bb6d-4bfd-b662-9f426950827f</wsa:MessageID>
          <wsa:RelatesTo>urn:uuid:fa161e28-bc73-4f92-8c47-815fdfc42677</wsa:RelatesTo>
          <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
          <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-bdac64c6-bb97-4287-93cf-6e90547619d5">
              <wsu:Created>2008-01-14T02:50:04Z</wsu:Created>
              <wsu:Expires>2008-01-14T02:55:04Z</wsu:Expires>
            </wsu:Timestamp>
          </wsse:Security>
        </soap:Header>
        <soap:Body>
          <PullFileResponse xmlns="http://htng.org/PWS/2008A/ODXService#PullFile">
            <PullFileResult>
              <TransferName>Test Transfer</TransferName>
              <TransferReference>http://htng.org/PWS/2008A/ODXService</TransferReference>
              <Status>
                <Code>Success</Code>
                <ErrorMessage>Your transmission was successful</ErrorMessage>
              </Status>
              <Files>
                <File>
                  <FileName>test.txt</FileName>
                  <SourceURI>A matching URI</SourceURI>
                  <DestinationURI>A matching URI</DestinationURI>

<File>SGkuICBIb3cgYXJlIHlvdSBkb2luZyB0b2RheT8gIEl0IGlzIGEgZ29vZCBkYXkgaXNuJ3QgaXQ/DQoNCkkgdGhpbmsgd2Ugc2hvdWxkIHRlc3QgYWxsIHRoZSBkYXRhL
g==</File>
                </File>
              </Files>
            </PullFileResult>
```

```
    </PullFileResponse>
  </soap:Body>
</soap:Envelope>
```

27.1.4 PullFile

A Pull File request message is submitted by a destination application to a source application in order to transfer one or more files from the source application to the destination application. MTOM (Message Transmission Optimization Mechanism) should be enabled.

Port	FileTransferPort
Operation	PullFile
Soap Action	http://htng.org/PWS/2008A/OpenDataExchange#PushFile
Input	HTNG_ODXPullFileRequest
Output	HTNG_ODXPullFileResponse
Primary Schema	FileTransfer.xsd
Role(s) Implemented	File Transfer Requestor, Fie Transfer Responder

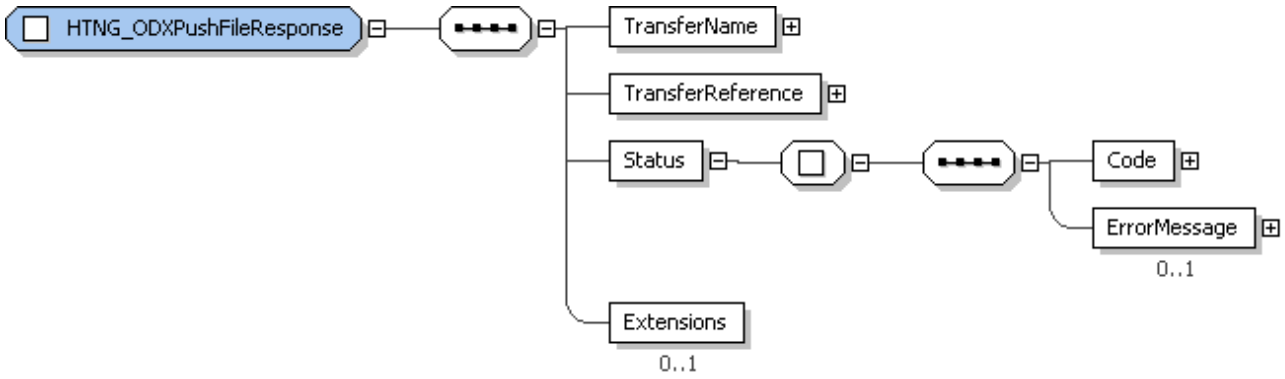


Element | @Attribute	Num	Description/Contents
HTNG_ODXPullFileRequest	1	Root element of the message
TransferName	1	Arbitrary name given to this transfer, should be meaningful to the responder.
TransferReference	1	Specific transfer reference, should be unique for each request message.
Description	0..1	Optional description.
Files	0..1	Optional element contains information about the files to be transfer. Optional as contents of TransferName may, through mutual agreement with the responder, dictate which file(s) are to be transferred.

File	1..N	Contains information about a file to be transferred.
FileName	1	The filename of the file being transferred.
SourceURI	0..1	Optional element containing the URI from where the file is sourced by the requestor.
DestinationURI	0..1	Optional element containing a URI that defines where the requestor should place the file.
Extensions	0..1	Optional element into which Requestor and Responder may place mutually agreed extensions.

PullFileRequest

```
<PullFile xmlns="http://htng.org/PWS/2008A/ODXService#PullFile">
      <Request>
        <filesField>
          <PullFileRequestFile>
            <FileName>string</FileName>
            <SourceURI>string</SourceURI>
            <DestinationURI>string</DestinationURI>
          </PullFileRequestFile>
          <PullFileRequestFile>
            <FileName>string</FileName>
            <SourceURI>string</SourceURI>
            <DestinationURI>string</DestinationURI>
          </PullFileRequestFile>
        </filesField>
        <TransferName>string</TransferName>
        <TransferReference>string</TransferReference>
        <Description>string</Description>
        <Files>
          <File>
            <FileName>string</FileName>
            <SourceURI>string</SourceURI>
            <DestinationURI>string</DestinationURI>
          </File>
          <File>
            <FileName>string</FileName>
            <SourceURI>string</SourceURI>
            <DestinationURI>string</DestinationURI>
          </File>
        </Files>
        <Extensions />
      </Request>
    </PullFile>
```



Element | @Attribute	Num	Description/Contents
HTNG_ODXPullFileResponse	1	Root element of the message
TransferName	1	Arbitrary name given to this transfer, should contain the same value as the request message.
TransferReference	1	Specific transfer reference, should contain the same value as the request message.
Status	1	Contains status information.
Code	1	Contains: 0 = Successful < 0: Error
ErrorMessage	0..1	If Code = Error, this element contains a message describing the error reason.
Files	0..1	Optional Element contains the files being transferred. Will not be present if an error occurred.
File	1..N	Contains a file being transferred.
FileName	1	The filename of the file being transferred.
FileImage	1	The file contents
SourceURI	0..1	Optional element containing the URI from where the file was sourced by the responder.
DestinationURI	0..1	Optional element containing a URI that defines where the requestor should place the file.
Extensions	0..1	Optional element into which Requestor and Responder may place mutually agreed extensions.

PullFileResponse	

	`<PullFileResponse xmlns="http://htng.org/PWS/2008A/ODXService#PullFile">`
	`  <PullFileResult>`
	`  <TransferName>string</TransferName>`
	`  <TransferReference>string</TransferReference>`
	`  <Status>`

```
          <Code>string</Code>
          <ErrorMessage>string</ErrorMessage>
        </Status>
        <Files>
          <File>
            <FileName>string</FileName>
            <FileImage>base64Binary</FileImage>
            <SourceURI>string</SourceURI>
            <DestinationURI>string</DestinationURI>
            <File>base64Binary</File>
          </File>
          <File>
            <FileName>string</FileName>
            <FileImage>base64Binary</FileImage>
            <SourceURI>string</SourceURI>
            <DestinationURI>string</DestinationURI>
            <File>base64Binary</File>
          </File>
        </Files>
        <Extensions />
      </PullFileResult>
    </PullFileResponse>
```

Sample Message
[*Request*]

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <soap:Header>
          <myHeader>
            <CorrelationID xmlns="http://htng.org/PWSWG/2007/02/AsyncHeaders">urn:uuid:882d75b9-f674-41ed-8cc2-
f0206fcbef18</CorrelationID>
            <ReplyTo xmlns="http://htng.org/PWSWG/2007/02/AsyncHeaders">
              <Address xmlns="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://localhost/ConsumerService/Service.asmx</Address>
            </ReplyTo>
            <FaultTo xmlns="http://htng.org/PWSWG/2007/02/AsyncHeaders">
              <Address xmlns="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://localhost/ConsumerService/Service.asmx</Address>
            </FaultTo>
          </myHeader>
          <wsa:Action>http://htng.org/PWS/2008A/ODXService#PushFile/PushFile</wsa:Action>
          <wsa:MessageID>urn:uuid:ebd6651d-058f-4ecb-80d1-dda226b9c05d</wsa:MessageID>
          <wsa:ReplyTo>
            <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
          </wsa:ReplyTo>
```

```
          <wsa:To>http://localhost:4060/ODXProviderService/ODXServices.asmx</wsa:To>
          <wsse:Security>
            <wsu:Timestamp wsu:Id="Timestamp-3e60948d-95c7-4788-8e53-f13a5a37e16b">
              <wsu:Created>2008-01-14T02:54:16Z</wsu:Created>
              <wsu:Expires>2008-01-14T02:59:16Z</wsu:Expires>
            </wsu:Timestamp>
          </wsse:Security>
        </soap:Header>
        <soap:Body>
          <PushFile xmlns="http://htng.org/PWS/2008A/ODXService#PushFile">
            <Request>
              <TransferName>Test Transfer</TransferName>
              <TransferReference>http://htng.org/PWS/2008A/ODXService</TransferReference>
              <Description>Simple push file test</Description>
              <Files>
                <File>
                  <FileName>test.txt</FileName>
                  <SourceURI>http://htng.org/PWS/2008A/ODXService/Consumer</SourceURI>
                  <DestinationURI>http://htng.org/PWS/2008A/ODXService/Producer</DestinationURI>
                  <File>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==</File>
                </File>
              </Files>
            </Request>
          </PushFile>
        </soap:Body>
      </soap:Envelope>
```


[*Response*]


```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <soap:Header>
          <myHeader>
            <CorrelationID xmlns="http://htng.org/PWSWG/2007/02/AsyncHeaders">urn:uuid:882d75b9-f674-41ed-8cc2-
f0206fcbef18</CorrelationID>
            <ReplyTo xmlns="http://htng.org/PWSWG/2007/02/AsyncHeaders">
              <Address xmlns="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://localhost/ConsumerService/Service.asmx</Address>
            </ReplyTo>
            <FaultTo xmlns="http://htng.org/PWSWG/2007/02/AsyncHeaders">
              <Address xmlns="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://localhost/ConsumerService/Service.asmx</Address>
            </FaultTo>
          </myHeader>
          <wsa:Action>http://htng.org/PWS/2008A/ODXService#PushFile/PushFileResponse</wsa:Action>
          <wsa:MessageID>urn:uuid:b465ff0f-6fa1-4e9d-8a49-4817d77ef019</wsa:MessageID>
          <wsa:RelatesTo>urn:uuid:ebd6651d-058f-4ecb-80d1-dda226b9c05d</wsa:RelatesTo>
          <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
          <wsse:Security>
```

```
          <wsu:Timestamp wsu:Id="Timestamp-6fbcea64-2bb5-41da-9f40-0a71e567a974">
            <wsu:Created>2008-01-14T02:54:16Z</wsu:Created>
            <wsu:Expires>2008-01-14T02:59:16Z</wsu:Expires>
          </wsu:Timestamp>
        </wsse:Security>
      </soap:Header>
      <soap:Body>
        <PushFileResponse xmlns="http://htng.org/PWS/2008A/ODXService#PushFile">
          <PushFileResult />
            <TransferName>Test Transfer</TransferName>
            <TransferReference>http://htng.org/PWS/2008A/ODXService</TransferReference>
            <Status>
              <Code>Success</Code>
              <ErrorMessage>Your transmission was successful</ErrorMessage>
            </Status>
        </PushFileResponse>
      </soap:Body>
    </soap:Envelope>
```

27.1.5 File Transfer XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with Oxygen XML Editor V8.2 by Paul Armstrong (MICROS Systems, Inc.) -->
<!-- updated with MS Visual Studio 2005 by Brian Alessi (Delaware North Companies, Inc.) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="FileTransfer"/>
  <xs:complexType name="PushFileRequest">
    <xs:sequence>
      <xs:element name="TransferName" type="xs:string" />
      <xs:element name="TransferReference" type="xs:string" />
      <xs:element name="Description" type="xs:string" minOccurs="0" />
      <xs:element name="Files">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="File" maxOccurs="unbounded">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="FileName" type="xs:string" />
                  <xs:element name="FileImage" type="xs:base64Binary" maxOccurs="1" />
                  <xs:element name="SourceURI" type="xs:string" minOccurs="0" />
                  <xs:element name="DestinationURI" type="xs:string" minOccurs="0" />
                  <xs:element name="File" type="xs:base64Binary" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
```

```
      </xs:complexType>
    </xs:element>
    <xs:element name="Extensions" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PushFileResponse">
  <xs:sequence>
    <xs:element name="TransferName" type="xs:string" />
    <xs:element name="TransferReference" type="xs:string" />
    <xs:element name="Status">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Code" type="xs:string" />
          <xs:element name="ErrorMessage" type="xs:string" minOccurs="0" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Extensions" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PullFileRequest">
  <xs:sequence>
    <xs:element name="TransferName" type="xs:string" />
    <xs:element name="TransferReference" type="xs:string" />
    <xs:element name="Description" type="xs:string" minOccurs="0" />
    <xs:element name="Files" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="File" maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="FileName" type="xs:string" />
                <xs:element name="SourceURI" type="xs:string" minOccurs="0" />
                <xs:element name="DestinationURI" type="xs:string" minOccurs="0" />
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Extensions" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PullFileResponse">
  <xs:sequence>
    <xs:element name="TransferName" type="xs:string" />
    <xs:element name="TransferReference" type="xs:string" />
    <xs:element name="Status">
```

```
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Code" type="xs:string" />
            <xs:element name="ErrorMessage" type="xs:string" minOccurs="0" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Files" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="File" maxOccurs="unbounded">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="FileName" type="xs:string" />
                  <xs:element name="FileImage" type="xs:base64Binary" minOccurs="1">
                  </xs:element>
                  <xs:element name="SourceURI" type="xs:string" minOccurs="0" />
                  <xs:element name="DestinationURI" type="xs:string" minOccurs="0" />
                  <xs:element name="File" type="xs:base64Binary" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Extensions" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Chapter 28 Implementation Requirements

28.1 Mandatory Requirements

A system must implement one or both of the File Transfer Requestor and File Transfer Responder roles. For the File Transfer Requestor role, the HTNG_ODXPushFileRequest and HTNG_ODXPullFileRequest messages must be implemented. For the File Transfer Responder role, the HTNG_ODXPushFileResponse and HTNG_ODXPullFileResponse messages must be implemented.

SOAP MTOM (Message Transmission Optimization Mechanism) must be enabled in the programming environment (e.g. .NET or J2EE) for both the Requestor and the Responder.

28.2 Implementation Options

The contents of the transferred file may also need to be encrypted to meet regulatory requirements for data security (such as Payment Card Industry Data Security Standards).

28.3 External Requirements

28.3.1 Mandatory Requirements

The HTNG Web Services Framework v2.1 must be implemented for transport of the messages. WS-Security must be supported, although it may be disabled in specific implementations where security is not an issue (such as transfer of files within a secure corporate network).

28.3.2 Implementation Options

SSL transport should be considered when transferring sensitive data.

Note: This specification is included for information and reference only. It is not included in the Certification Program at this time. It is subject to change in the future.

Part 5 Back Office Integration Version 1.4

Chapter 29 Document History

29.1 Document Changes

29.1.1 Integration Specification

Version	Date	Author	Comments
1.0	July 19, 2007	Dieter Krause SAP AG	Initial submission
1.1	August 3, 2007	Dieter Krause SAP AG	GetMapping and CheckMapping included
1.2	August 10,2007	Dieter Krause SAP AG	PostGL: single side transactions allowed. Attribute “TransactionMode” added CheckMapping: merged with GetMapping. GetMapping results extended with dimensions and debit/credit information for accounts Whole document: name of element “Standarddimensions” changed to “Dimension” , since the set of dimensions is not restricted by the specification
1.3	August 17, 2007	Dieter Krause SAP AG	GetMapping and PostAR modified
1.4	August 30, 2007	Dieter Krause SAP AG	MessageNumber added to header, PostGL,PostAR and PostStatistics modified (e.g. ServiceDate added)
1.4	December 19, 2007	Dieter Krause SAP AG	Final editing

29.1.2 Use Cases

Version	Date	Author	Comments
0.9	June 1, 2007	Dieter Krause SAP AG	Initial proposal, incomplete, for format discussion only
1.0	July 26, 2007	Dieter Krause SAP AG	More Use cases added
1.1	August 2,2007	Dieter Krause SAP AG	Reflects comments by Micros
1.2	August 10,2007	Dieter Krause SAP AG	Corresponds to Specification Document Version 1.2
1.3	August 17, 2007	Dieter Krause SAP AG	Use cases 9 -11 added (PostAR applied to groups) , changes in specification document reflected
1.4	August 23, 2007	Dieter Krause SAP AG	use cases adapted to include MessageNumber
1.4	December 19, 2007	Dieter Krause SAP AG	Final editing

29.1.3 Message Specification

Version	Date	Author	Comments
	January 26, 2006	Tom Gresham, MICROS Systems, Inc.	Initial submission
	August 16, 2007	Tom Gresham, MICROS Systems, Inc.	Revision per workgroup discussions
	August 20, 2007	Tom Gresham, MICROS Systems, Inc.	Accounts receivable and mapping sections
	August 29, 2007	Tom Gresham, MICROS Systems, Inc.	Correct statistical posting message and additions to accounts receivable message
	September 11, 2007	Tom Gresham, MICROS Systems, Inc.	Add missing Message Number to transaction header and update namespace to Sept. 2007
	September 20, 2007	Tom Gresham, MICROS Systems, Inc.	Change mapping detail to include multiple dimensions on source element.

29.1.4 Integrated Document

Version	Date	Author	Comments
1.5	January 15, 2007	Martin Kirk, The Open Group	Merge to create final document

Chapter 30 Acknowledgements

30.1 Contributors

30.1.1 Message Definitions

Company / Organization	Contributor
Delaware North Companies	Yvette Vincent Brian Fending Brian Alessi
Epicor Software Corporation	Sander Bremer Adam Prince
Folio Exchange	Glen Miller
Four Seasons Hotels & Resorts	Natasha Nelson
Global Hyatt Corp.	Tom Pascarella
HTNG	Douglas Rice
Infor	Lee Coogan John Beall
MICROS Systems, Inc.	Tom Gresham
PAR Springer-Miller Systems	Brigitte Guy Penka Sevova
SAP	Axel Janz Dieter Krause
TAC GmbH	Thomas Rössler Bernhard Rappold
Delaware North Companies	Yvette Vincent Brian Fending Brian Alessi

Chapter 31 Document Information

31.1 Document Purpose

This document describes a Back Office Interface for the hotel industry. It also provides functional business and technical descriptions of the web services required. Expertise in XML, SOAP and HTTP required.
The document also includes extensive Use cases.

31.2 Document Terms

For the purpose of this document the following terms have been defined as follows:

Term	Definition
RCS	Revenue Capturing Systems (e.g. a PMS): generate transactions records from different activities of a hotel property which are to be passed to the accounting system either in detail form or as summary information
Mapping Layer	Software layer, which transforms values used by the RCS into values used by the accounting system. Example: transaction codes used by the revenue capturing system must be mapped to accounts in the accounting system using various parameters
Accounting System	System which stores summary or detailed accounting information sent from revenue capturing systems for bookkeeping or statistical purposes

31.3 Referenced Documents

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

Name	Location
HTNG Web Service Framework 2.0	Describes the web services framework standards that implementers must adhere to

Chapter 32 Business Process

This document defines a generic message format for exchange of accounting data between a revenue capturing system and the back office system in a hotel environment, and this section shows by examples how these services can be used in practice to model different business events (called "Use Cases") in a hotel or related institutions like a Spa, and how the resulting postings in the accounting system could look like.

It has be understood, that these postings are only examples and could be done differently in different countries due to legislation or common practice.

32.1 Use Cases

32.1.1 Post Revenues

Use Case ID	**PMS01**
Name	Post revenues
Brief description	This use case describes the case when the RCS (Revenue Capturing System) is sending revenue data to the Accounting system, split by reservation. The RCS is a PMS system. It is sending a revenue of EUR 100 net for room charges. The revenues are qualified by several dimensions.
Provider of service	Accounting System
Actor	Revenue generating system

Resulting Accounts:

Guest Ledger

Debit	Credit
100.00	

Room Charge

Debit	Credit
	100.00

```
<PostGl>
<Header TransactionMode="SingleSide'>
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<TransactionDate>4-DEC-06</TransactionDate>
<ServiceDate>4-DEC-06</ServiceDate>
<SourceSystemType>PMS</SourceSystemType>
```

```
<SourceSystemId>PMS0001</SourceSystemId>
<Reservation#>4711</Reservation#>
</Header>

<Transaction>
    <Credit>
<TransactionCode>RoomChargeNet</TransactionCode>
< Amount>100</ Amount>
< Currency>EUR</ Currency>
<TransactionText> Room Charge 4-DEC-06</ TransactionText>
    </Credit>

  <StandardDimension>
<DimensionName>RM_Type</DimensionName>
<DimensionValue>Standard</DimensionValue>
</StandardDimension>
<StandardDimension>
<DimensionName>Market</DimensionName>
      <DimensionValue>Group</DimensionValue>
</StandardDimension>
</Transaction>
</PostGl>
```

Use Case ID	**PMS01a**
Name	Post revenues
Brief description	This use case describes the case when the RCS (Revenue Capturing System) is sending revenue data to the Accounting system In this case, the RCS is a Spa system. Its is sending a revenue for BodyTreatment of EUR 100 net. The revenues are qualified by several dimensions:
Provider of service	Accounting System
Actor	Revenue generating system

Resulting Accounts:

Guest Ledger

Debit	Credit
100.00	

RevenueBodyTreatment

Debit	Credit
	100.00

```
<PostGl>
<Header TransactionMode='SingleSide">
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<TransactionDate>4-DEC-06</TransactionDate>
<SourceSystemType>SPA</SourceSystemType>
<SourceSystemId>SPA0001</SourceSystemId>
</Header>

<Transaction>
    <Credit>
<TransactionCode>RevenueBodyTreatment</TransactionCode>
< Amount>100</ Amount>
< Currency>EUR</ Currency>
<TransactionText> Room Charge 4-DEC-06</ TransactionText>
    </Credit>

  <StandardDimension>
<DimensionName>FacilityType</DimensionName>
<DimensionValue>Pool</DimensionValue>
</StandardDimension>
<StandardDimension>
<DimensionName>ProviderType</DimensionName>
      <DimensionValue>Internal </DimensionValue>
</StandardDimension>
</Transaction>
</PostGl>
```

Use Case ID	**PMS01b**
Name	Post revenues
Brief description	This use case describes the case when the RCS (Revenue Capturing System) is sending revenue data to the Accounting system, split by reservation, **including tax posting**. The RCS is a PMS system. Its is sending a revenue of USD 100 net for room charges + USD for 10 Tax payable The revenues are qualified by several dimensions.
Provider of service	Accounting System
Actor	Revenue generating system

Resulting Accounts:

Guest Ledger

Debit	Credit
110.00	

Room Charge

Debit	Credit
	100.00

TaxPayable

Debit	Credit
	10

```
<PostGl>
<Header TransactionMode='SingleSide">
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<TransactionDate>4-DEC-06</TrandsactionDate>
<SourceSystemType>PMS</SourceSystemType>
<SourceSystemId>PMS0001</SourceSystemId>
<Reservation#>4711</Reservation#>
</Header>

<Transaction>
    <Credit>
<TransactionCode>RoomChargeNet</TransactionCode>
< Amount>100</ Amount>
< Currency>USD</ Currency>
<TransactionText> Room Charge 4-DEC-06</ TransactionText>
    </Credit>

  <StandardDimension>
<DimensionName>RM_Type</DimensionName>
<DimensionValue>Standard</DimensionValue>
</StandardDimension>
<StandardDimension>
<DimensionName>Market</DimensionName>
      <DimensionValue>Group</DimensionValue>
</StandardDimension>
</Transaction>
<Transaction>
  <Credit>
<TransactionCode>TaxPayable</TransactionCode>
< Amount>10</ Amount>
< Currency>USD</ Currency>
<TransactionText> Room Charge 4-DEC-06</ TransactionText>
    </Credit
```

```
  </Transaction>
</PostGl>
```

32.1.2 Post a Payment at Checkout

Use Case ID	**PMS02**
Name	Post a payment at check out
Brief description	This use case describes the case when the RCS (Revenue capturing system) is sending payment data to the accounting system. The payment has been received from a guest by VisaCard. Payment of USD 110 at Check out by Visa
Provider of Service	Accounting System
Actor	Revenue capturing system

Resulting Accounts:

Guest Ledger

Debit	Credit
	100

Visa

Debit	Credit
110	

TaxVAT

Debit	Credit
	10

```
<PostGl>
<Header TransactionMode='SingleSide">
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<TransactionDate>4-DEC-06</TransactionDate>
<SourceSystemType>PMS</SourceSystemType>
<SourceSystemId>PMS0001</SourceSystemId>
<CustomerCode>P411</CustomerCode>
</Header>
<Transaction>
    <Debit>
<TransactionCode>PaymentVisa</TransactionCode>
< Amount>110</ Amount>
< Currency>USD</ Currency>
```

```
<TransactionText> Room Charge 4-DEC-06</ TransactionText>
     </Debit>
    </Transaction>
  </PostGl>
```

32.1.3 Move from Deposit Ledger to Guest Ledger

Use Case ID	**PMS03**
Name	Move from Deposit Ledger to Guest Ledger
Brief description	This use case describes the following case: A guest has paid a deposit. When he checks in, the deposit is moved to the guest ledger.
Provider of Service	Accounting System
Actor	Revenue capturing system

Resulting Accounts:

Deposit Ledger

Debit	Credit
100 (1)	

Guest Ledger

Debit	Credit
	100 (1)

```
<PostGl>
<Header TransactionMode='SingleSide">
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<TransactionDate>4-DEC-06</TransactionDate>
<SourceSystemType>PMS</SourceSystemType>
<SourceSystemId>PMS0001</SourceSystemId>
<CustomerCode>P411</CustomerCode>
</Header>
<Transaction>
    <Debit>
           <TransactionCode>MoveDeposit </TransactionCode>
<Amount>100</LedgerAmount>
<Currency>USD</Currency>
    </Debit>
      </Transaction>
</PostGl>
```

32.1.4 Transfer of Statistical Data

Use Case ID	**PMS04**
Name	Transfer of Statistical Data
Brief description	The RCS sends the number of available rooms for a specific day for room type standard and market segment Rack
Provider of Service	Accounting System
Actor	Revenue capturing system

```
<PostStatistics>
<Header>
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<TransactionDate>4-DEC-06</TransactionDate>
<SourceSystemType>PMS</SourceSystemType>
<SourceSystemId>PMS0001</SourceSystemId>
<User>JMiller</User>
</Header>
<Statistic>
< StatisticalData>
    <Description >AvailableRooms  </ Description >
     < Value> 101 </ Value>
</StatisticalData>
< Dimensions>
<DimensionName>Roomtype</DimensionName>
<DimensionValue>Standard</DimensionValue>
</ Dimensions>
< Dimensions>
<DimensionName>MarketSegment</DimensionName>
<DimensionValue>Rack</DimensionValue>
</ Dimensions>
<Statistic>
</PostStatistics>
```

32.1.5 Deposit

Use Case ID	**PMS05**
Name	Deposit
Brief description	Deposit of USD1000.00 with credit card Visa for Reservation 4711 by customer P-000111 . Mapping is done by RCS
Provider	Accounting System
Actor	Revenue capturing system

Resulting Accounts :

Deposit ledger

Debit	Credit
	1000

Visa card

Debit	Credit
1000	

```
<PostGl>
<Header  TransactionMode='PostingRecord'>
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<TransactionDate>4-DEC-06</TransactionDate>
<SourceSystemType>PMS</SourceSystemType>
<SourceSystemId>PMS0001</SourceSystemId>
 <Reservation#>4711</Reservation#>
<CustomerCode>P-000111</CustomerCode>
</Header>
<Transaction>
    <Credit>
            < TransactionAccount Number >100000 </ TransactionAccount Number >
<Amount>1000</LedgerAmount>
<Currency>USD</Currency>
    </Credit>
    <Debit>
             < TransactionAccount Number >210000</ TransactionAccount Number >
<Amount>1000</ Amount>
<Currency>USD</Currency>
  </Debit>
  </Transaction>
</PostGl>
```

32.1.6 Guest Invoice with Discount

Use Case ID	**PMS06**
Name	Guest Invoice with discount
Brief description	A guest stays one night in Standard Room (100.00) with invoice and a discount of 10%. Reservation# is 4711. Tax is 10% Two messages are used by the RCS: The first message posts the revenue + discount The second message posts the invoice + Tax

Provider	Accounting System
Actor	Revenue capturing system

Message 1 (Night Audit):

```
<PostGl>
<Header TransactionMode='SingleSide'>
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<TransactionDate>4-DEC-06</TransactionDate>
<SourceSystemType>PMS</SourceSystemType>
<SourceSystemId>PMS0001</SourceSystemId>
<Reservation#>4711</Reservation#>
</Header>

<Transaction>
    <Credit>
<TransactionCode>RoomCharge </TransactionCode>
< Amount>100</ Amount>
< Currency>USD</ Currency>
<TransactionText> Room Charge 4-DEC-06</ TransactionText>
    </Credit>
  <StandardDimension>
<DimensionName>RM_Type</DimensionName>
<DimensionValue>Standard</DimensionValue>
</StandardDimension>
<StandardDimension>
<DimensionName>Market</DimensionName>
      <DimensionValue>Group</DimensionValue>
</StandardDimension>
</Transaction>
  </PostGl>
```

Message 2 (Check out)

```
<PostGl>
<Header TransactionMode='SingleSide'>
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<Date>6-DEC-06</Date>
<SourceSystemType>PMS</SourceSystemType>
<SourceSystemId>PMS0001</SourceSystemId>
<Reservation#>4711</Reservation#>
</Header>

<Transaction>
    <Debit>
```

```
<TransactionCode>Discount</TransactionCode>
< Amount>10</ Amount>
< Currency>USD</ Currency>
<TransactionText> Room Charge 4-DEC-06</ TransactionText>
    </Debit>
    <StandardDimension>
<DimensionName>RM_Type</DimensionName>
<DimensionValue>Standard</DimensionValue>
</StandardDimension>
<StandardDimension>
<DimensionName>Market</DimensionName>
      <DimensionValue>Group</DimensionValue>
</StandardDimension>
</Transaction>


</PostGl>
```

Resulting Accounts :

Guest Ledger

Debit	Credit
100.00 (1)	10 (2)

Room Charge

Debit	Credit
	100.00 (1)

Discount

Debit	Credit
10 (2)	

Message 3 (Check Out):

```
<PostAR>
<Header>
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<Date>6-DEC-06</Date>
<SourceSystemType>PMS</SourceSystemType>
```

```
<SourceSystemId>PMS0001</SourceSystemId>
<Reservation#>4711</Reservation#>
</Header>
<InvoiceNumber>A1111</InvoiceNumber>
<InvoiceDate>6-DEC-06</InvoiceDate>
<DueDate>26-DEC-06</DueDate>
<TotalInvoiceAmount>99<TotalInvoiceAmount>
<Currency>USD</Currency>
<Tax>
<Taxrate>A2</TaxRate>
<TaxBase>90</TaxBase>
<TaxAmount>9 </TaxAmount>
<TaxIncluded>FALSE</TaxIncluded>
</Tax>
<Text>Invoice  A1111</Text>
</PostAR>
```

Resulting Accounts :

Guest Ledger

Debit	Credit	Reference
	90 (1)	P-000111

CityLedger or A/R

Debit	Credit	Reference
99 (1)		P-000111

TaxA2

Debit	Credit	Reference
	9	P-000111

32.1.7 Partial Payment from F&B Point of Sale System

Use Case ID	**PMS07**
Name	Partial Payment from F&B Point of Sale System
Brief description	Two guests go have lunch at a Casual restaurant in Hotel "HTNG" At the end of the meal, the check is presented to them. Instead of figuring out who is paying for what, guests decide to just split this check 50/50 They ask the server to perform a 50/50 split in following manner 50% to a Visa card of one guest

	50% to the guestroom of one of the guests
Provider	Accounting System
Actor	Revenue capturing system

1. Restaurant Settlement

- Two guests go have lunch at a Casual restaurant in Hotel "HTNG"
- At the end of the meal, the following check is presented to them:

Quantity	Item	Price	Total
1	Caesar Salad	10.00	10.00
1	House Salad	8.00	8.00
1	Sandwich	15.00	15.00
1	Pasta	17.00	17.00
2	Lemonade	3.00	6.00
	Food Tax (10%)	5.00	5.00
	Beverage Tax (15%)	0.90	0.90
	Gratuity	10.00	10.00
	TOTAL		**71.90**

- Instead of figuring out who is paying for what, guests decide to just split this check 50/50
- They ask the server to perform a 50/50 split in following manner
    - 50% to a Visa card of one guest
    - 50% to the guestroom of one of the guests
- The following settled check is then presented to them

Quantity	Item	Price	Total
1	Caesar Salad	10.00	10.00
1	House Salad	8.00	8.00
1	Sandwich	15.00	15.00
1	Pasta	17.00	17.00
2	Lemonade	3.00	6.00
	Tax	5.90	5.90
	Gratuity	10.00	10.00
	TOTAL		**71.90**
Payment	**Visa**		**(35.95)**
	Room Charge (Mr. 1001)		**(35.95)**

2. POS – PMS Interface procedure

- Now, the POS system needs to send Room Charge Over to the PMS so that the charge can be posted to the guest folio
- For the purposes of Package Splitting and third party billing flexibility, the requirement for the POS/PMS interface is to split Room Charges by Itemizers by Meal Period, by Restaurant
- The following Mapping is setup in the POS/PMS Interface

RVC#	RVC Name	Itemizers	Serving Period			
			Breakfast	**Lunch**	**Dinner**	**Brunch**
	Casual Restaurant	Food	**2600**	**2620**	**2640**	**2660**
		Beverage	**2601**	**2621**	**2641**	**2661**
		Masc.	**2605**	**2625**	**2645**	**2665**
		SPLIT	**2600**	**2620**	**2640**	**2660**
		Gratuity	**2608**	**2628**	**2648**	**2668**
		Tax	**2609**	**2629**	**2649**	**2669**

- The codes 2600, 2601, etc represent PMS Transaction Codes
- SPLIT itemizer is used to auto-balance Partial Payment to the Room Charge
- Typically, SPLIT is mapped to Food of the corresponding Meal Period. While this does not correctly represent the reality, the ambiguity of partial payment has to be addressed somehow, since guests have not decided exactly which items they would want to go towards who's payment
- Given this mapping, the check is sent to PMS, onto Guest Folio in room 1001, in the following fashion:

Quantity	Item	Price	Total	PMS Trans Code
1	Caesar Salad	10.00	10.00	2620
1	House Salad	8.00	8.00	2620
1	Sandwich	15.00	15.00	2620
1	Pasta	17.00	17.00	2620
2	Lemonade	3.00	6.00	2621
	Food Tax (10%)	5.00	5.00	2629
	Beverage Tax (15%)	0.90	0.90	2629
	Gratuity	10.00	10.00	2628
	SPLIT		(35.95)	2620

- It now appears on the guest folio in PMS as follows:

Casual Restaurant Lunch	35.95

- Behind the scenes, same folio is represented as follows:

Code	Item	Amount
2620	Casual Restaurant Food	14.05
2621	Casual Restaurant Beverage	6.00

2628	Casual Restaurant Tax	5.90
2629	Casual Restaurant Gratuity	10.00

3. PMS to Back Office Interface Procedure

- Given that we now have POS Revenues interfaced directly to Back Office, it's no longer necessary to rely on PMS revenues split for Food and Beverage to report F&B Revenue. The reason we have it split in PMS into Food, Beverage, Tax, Gratuity is to allow PMS to allow to bill certain items only. For example, if a third party requests to pay for one's food, but will not cover their beverage. Or, if a package includes Food and Beverage, but guests have to pay their own tax and gratuity
- Since POS is the primary source of Revenue reporting for F&B, we can simply map all of the F&B accounts to one clearing G/L account in Back Office (perhaps one for each restaurant).
- At the same time, POS will map Room Charge Payment to the same Clearing Account to offset the amounts.
- Therefore, in the mapping system, the following mapping will be setup:

PMS Trans Code	Description	Back Office G/L Code
2620	Casual Restaurant Food	200-100
2621	Casual Restaurant Beverage	200-100
2628	Casual Restaurant Tax	200-100
2629	Casual Restaurant Gratuity	200-100

- In this instance, the following transactions will be posted to the G/L

Debit	Credit	Account
35.95		Guest Ledger
	35.95	200-100

Assuming, that the mapping has already been done in the PMS, the resulting message is the following:

```
<PostGl>
<Header TransactionMode = 'SingleSide'>
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<TransactionDate>6-DEC-06</TransactionDate>
<SourceSystemType>PMS</SourceSystemType>
<SourceSystemId>PMS0001</SourceSystemId>
 </Header>

<Transaction>
    <Credit>
<TransactionAccountNmber>200-100</ TransactionAccountNmber >
< Amount>35.95</ Amount>
< Currency>USD</ Currency>
<TransactionText> Room Charge 4-DEC-06</ TransactionText>
    </Credit>
    </Transaction>
```

```
</PostGl
```

4. POS to Back Office Mapping

- The following mapping exists between POS and Back Office system:

RVC Name	Itemizers	BackOffice G/L Code
Casual Restaurant	Food	200-201
Casual Restaurant	Beverage	200-202
Casual Restaurant	Misc	200-203
Casual Restaurant	Gratuity	200-204
Casual Restaurant	Tax	200-205
Casual Restaurant	Room Charge Settlement	200-100
Casual Restaurant	Visa Settlement	200-101

- Therefore, the afore mentioned check would interface to a Back Office system in the following manner

Debit	Credit	Account
35.95		200-100
35.95		200-101
	50.00	200-201
	6.00	200-202
	10.00	200-204
	5.90	200-205

Assuming, that the mapping has already been done in the PMS, the resulting message is the following:

```
<PostGl>
<Header TransactionMode = 'PostingRecord'>
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<TransactionDate>6-DEC-06</TransactionDate>
<SourceSystemType>F&B</SourceSystemType>
<SourceSystemId>POS0001</SourceSystemId>
 </Header>

<Transaction>
    <Debit>
<TransactionAccountNumber>200-100</ TransactionAccountNumber >
< Amount>35.95</ Amount>
< Currency>USD</ Currency>
<TransactionText> Offset for Partial Payment</ TransactionText>
     </Debit>
    <Debit>
<TransactionAccountNumber>200-101</ TransactionAccountNumber >
```

```
< Amount>35.95</ Amount>
< Currency>USD</ Currency>
<TransactionText> VisaCard Payment </ TransactionText>
     </Debit>

    <Credit>
<TransactionAccountNumber>200-201</ TransactionAccountNumber >
< Amount>50.00</ Amount>
< Currency>USD</ Currency>
<TransactionText> RevenueFood </ TransactionText>
    </Credit >
<Credit>
<TransactionAccountNumber>200-202</ TransactionAccountNumber >
< Amount>6.00</ Amount>
< Currency>USD</ Currency>
<TransactionText> RevenueBeverage </ TransactionText>
    </Credit >
  <Credit>
<TransactionAccountNumber>200-204</ TransactionAccountNmber >
< Amount>10.00</ Amount>
< Currency>USD</ Currency>
<TransactionText> RevenueGratuity</ TransactionText>
    </Credit >
   <Credit>
<TransactionAccountNumber>200-205</ TransactionAccountNumber >
< Amount>5.90</ Amount>
< Currency>USD</ Currency>
<TransactionText> Tax</ TransactionText>
    </Credit
</Transaction>
</PostGl>
```

5. Combined posting of the same guest check from PMS and POS into Back Office System:

Debit	**Credit**	**Account**
35.95		Guest Ledger
	35.95	**200-100**
35.95		**200-100**
35.95		200-101
	50.00	200-201
	6.00	200-202
	10.00	200-204
	5.90	200-205

As we can see from this examination, the revenues have been reported correctly, and the clearing account 200-100 has received offsetting postings.

32.1.8 Correction of Revenue Posting

Use Case ID	**PMS08**
Name	Correction of revenue postings
Brief description	This use case describes the case when the RCS (Revenue Capturing System) has to correct postings which have been done earlier, i.e. during precedent night audits This could be triggered for example by a guest checking his invoice and complaining about wrong beverage charges included in the invoice.
Provider of service	Accounting System
Actor	Revenue generating system

Variant 1:

Resulting Accounts:

Guest Ledger

Debit	Credit
	100

Corrections

Debit	Credit
100	

```
<PostGl>
<Header TransactionMode = 'SingleSide'>
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<TransactionDate>4-DEC-06</TransactionDate>
<SourceSystemType>PMS</SourceSystemType>
<SourceSystemId>PMS0001</SourceSystemId>
<Reservation#>4711</Reservation#>
</Header>

<Transaction>
    <Credit>
<TransactionCode>Beverage</TransactionCode>
< Amount> -100 </ Amount>
< Currency>EUR</ Currency>
<TransactionText> Room Charge 4-DEC-06</ TransactionText>
    </Credit>
    <StandardDimension>
<DimensionName>RM_Type</DimensionName>
```

```
<DimensionValue>Standard</DimensionValue>
</StandardDimension>
<StandardDimension>
<DimensionName>Market</DimensionName>
      <DimensionValue>Group</DimensionValue>
</StandardDimension >
</Transaction>
</PostGl>
```

Variant 2:

Resulting Accounts:

Guest Ledger

Debit	Credit
-100	

Beverage

Debit	Credit
	-100

```
<PostGl>
<Header TransactionMode = 'SingleSide'>
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<TransactionDate>4-DEC-06</TransactionDate>
<SourceSystemType>PMS</SourceSystemType>
<SourceSystemId>PMS0001</SourceSystemId>
<Reservation#>4711</Reservation#>
</Header>

<Transaction>
    <Debit>
<TransactionCode>Correction</TransactionCode>
< Amount>100</ Amount>
< Currency>EUR</ Currency>
<TransactionText> Room Charge 4-DEC-06</ TransactionText>
    </Debit>
<StandardDimension>
<DimensionName>RM_Type</DimensionName>
<DimensionValue>Standard</DimensionValue>
</StandardDimension>
```

```
<StandardDimension>
<DimensionName>Market</DimensionName>
      <DimensionValue>Group</DimensionValue>
</StandardDimension>
</Transaction>
</PostGl>
```

32.1.9 Guest Invoice with Line Detail Billed to 3rd party to City Ledger

Use Case ID	**PMS09**
Name	Guest invoice with line detail billed to 3rd party to city ledger
Brief description	A guest stays two nights in Standard Room (100/night) with 20 porterage charge. Invoice is sent to a wholesaler whose reference number (booking / trip / tour / etc.) is 123456. (Note: taxes were applied before transfer from Guest Ledger.) One message is used by the RCS: The first message posts the invoice. Billing is done by RCS and includes line detail.
Provider	Revenue Capturing System
Actor	Accounting System

Message 1 (Check Out):

```
<PostAR>
<Header>
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<Date>6-DEC-06</Date>
<SourceSystemType>PMS</SourceSystemType>
<SourceSystemId>PMS0001</SourceSystemId>
<CustomerCode>ABC</CustomerCode>
</Header>
<InvoiceNumber>A1111</InvoiceNumber>
<InvoiceDate>6-DEC-06</InvoiceDate>
<DueDate>26-DEC-06</DueDate>
<TotalInvoiceAmount>220<TotalInvoiceAmount>
<Currency>USD</Currency>

<StandardDimension>
<DimensionName>CustomerReference</DimensionName>
<DimensionValue>123456</DimensionValue>
</StandardDimension>

<Text>Smith, Steve and Susan</Text>

<InvoiceLine>
```

```
<LineNumber>1</LineNumber>
<FolioNumber>1</FolioNumber>
<Text>Porterage</Text>
<Amount>19</Amount>
<TransactionDate>4-DEC-06</TransactionDate>
</InvoiceLine>

<InvoiceLine>
<LineNumber>2</LineNumber>
<FolioNumber>1</FolioNumber>
<Text>Excise Tax</Text>
<Amount>1</Amount>
<TransactionDate>4-DEC-06</TransactionDate>
</InvoiceLine>

<InvoiceLine>
<LineNumber>3</LineNumber>
<FolioNumber>1</FolioNumber>
<Text>Room Charge</Text>
<Amount>90</Amount>
<TransactionDate>5-DEC-06</TransactionDate>
</InvoiceLine>

<InvoiceLine>
<LineNumber>4</LineNumber>
<FolioNumber>1</FolioNumber>
<Text>Occupancy Tax</Text>
<Amount>10</Amount>
<TransactionDate>5-DEC-06</TransactionDate>
</InvoiceLine>

<InvoiceLine>
<LineNumber>5</LineNumber>
<FolioNumber>1</FolioNumber>
<Text>Room Charge</Text>
<Amount>90</Amount>
<TransactionDate>6-DEC-06</TransactionDate>
</InvoiceLine>

<InvoiceLine>
<LineNumber>6</LineNumber>
<FolioNumber>1</FolioNumber>
<Text>Occupancy Tax</Text>
<Amount>10</Amount>
<TransactionDate>6-DEC-06</TransactionDate>
</InvoiceLine>
</PostAR>
```

Resulting Accounts :

Guest Ledger

Debit	Credit
	220 (1)

CityLedger or A/R

Debit	Credit
220 (1)	

32.1.10 Group Invoice Billed to 3rd Party to City Ledger with Group Detail

Use Case ID	**PMS10**
Name	Group invoice billed to 3rd party to city ledger with group detail
Brief description	A group belonging to a corporate account consists of two rooms for one night (100 x 2) and a banquet charge (200) checks out. Guest folios are transferred to the Group Folio (a.k.a., Posting Master). Invoice is sent to a corporate account whose reference number (booking / trip / tour / etc.) is A007. Taxes have been applied before transfer to City Ledger. One message is used by the RCS: The first message posts the invoice Billing is done by RCS and includes line detail (but not room folio level detail for transferred guest folios).
Provider	Revenue Capturing System
Actor	Accounting System

Message 1 (Group / Posting Master Check Out):

```
<PostInvoice>
<Header>
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<Date>6-DEC-06</Date>
<SourceSystemType>PMS</SourceSystemType>
<SourceSystemId>PMS0001</SourceSystemId>
<CustomerCode>BCD</CustomerCode>
</Header>
<InvoiceNumber>A1112</InvoiceNumber>
<InvoiceDate>6-DEC-06</InvoiceDate>
<DueDate>26-DEC-06</DueDate>
<TotalInvoiceAmount>400<TotalInvoiceAmount>
<Currency>USD</Currency>

<StandardDimension>
<DimensionName>GroupId</DimensionName>
<DimensionValue>P0011</DimensionValue>
</StandardDimension>

<StandardDimension>
<DimensionName>CustomerReference</DimensionName>
<DimensionValue>A007</DimensionValue>
</StandardDimension>

<Text>BigCo Rewards Celebration</Text>

<InvoiceLine>
```

```
<LineNumber>1</LineNumber>
<Text>Banquet Charges</Text>
<Amount>160</Amount>
<TransactionDate>5-DEC-06</TransactionDate>
</InvoiceLine>

<InvoiceLine>
<LineNumber>2</LineNumber>
<Text>Dining Tax</Text>
<Amount>40</Amount>
<TransactionDate>5-DEC-06</TransactionDate>
</InvoiceLine>

<InvoiceLine>
<LineNumber>3</LineNumber>
<Text>XFer from Sue Matsumoto</Text>
<Amount>100</Amount>
<TransactionDate>6-DEC-06</TransactionDate>
</InvoiceLine>

<InvoiceLine>
<LineNumber>4</LineNumber>
<Text>XFer from Charles Smith</Text>
<Amount>100</Amount>
<TransactionDate>6-DEC-06</TransactionDate>
</InvoiceLine>

</PostInvoice>
```

Resulting Accounts :

Guest Ledger

Debit	Credit
	400 (1)

CityLedger or A/R

Debit	Credit
400 (1)	

32.1.11 Group Invoice Billed to 3rd Party to City Ledger with Line Charge Detail

Use Case ID	**PMS11**

Name	Group invoice billed to 3rd party to city ledger with line charge detail
Brief description	A group belonging to a corporate account consists of two rooms for two nights (200 x 2) and a banquet charge (200) checks out. Guest folios are transferred to the Group Folio (a.k.a., Posting Master). Invoice is sent to a corporate account whose reference number (booking / trip / tour / etc.) is A007. Taxes have been applied before transfer to City Ledger. One message is used by the RCS: The first message posts the invoice Billing is done by RCS and includes line detail including room folio level detail for transferred guest folios.
Provider	Revenue Capturing System
Actor	Accounting System

Message 1 (Group / Posting Master Check Out):

```
<PostAR>
<Header>
<MessageNumber> ABC12345</MessageNumber>
<PropertyCode>01</PropertyCode>
<Date>6-DEC-06</Date>
<SourceSystemType>PMS</SourceSystemType>
<SourceSystemId>PMS0001</SourceSystemId>
<CustomerCode>BCD</CustomerCode>
</Header>
<InvoiceNumber>A1112</InvoiceNumber>
<InvoiceDate>6-DEC-06</InvoiceDate>
<DueDate>26-DEC-06</DueDate>
<TotalInvoiceAmount>600<TotalInvoiceAmount>
<Currency>USD</Currency>

<StandardDimension>
<DimensionName>GroupId</DimensionName>
<DimensionValue>P0011</DimensionValue>
</StandardDimension>

<StandardDimension>
<DimensionName>CustomerReference</DimensionName>
<DimensionValue>A007</DimensionValue>
</StandardDimension>

<Text>BigCo Rewards Celebration</Text>

<InvoiceLine>
<LineNumber>1</LineNumber>
<Text>Banquet Charges</Text>
<Amount>160</Amount>
<TransactionDate>5-DEC-06</TransactionDate>
```

```
</InvoiceLine>

<InvoiceLine>
<LineNumber>2</LineNumber>
<Text>Dining Tax</Text>
<Amount>40</Amount>
<TransactionDate>5-DEC-06</TransactionDate>
</InvoiceLine>

<InvoiceLine>
<LineNumber>3</LineNumber>
<Text>XFer from Sue Matsumoto</Text>
<Amount>200</Amount>
<TransactionDate>6-DEC-06</TransactionDate>

<InvoiceLine>
   <LineNumber>1</LineNumber>
<Text>Room Charges</Text>
<Amount>90</Amount>
<TransactionDate>5-DEC-06</TransactionDate>
</InvoiceLine>

<InvoiceLine>
   <LineNumber>2</LineNumber>
<Text>Taxes</Text>
<Amount>10</Amount>
<TransactionDate>5-DEC-06</TransactionDate>
</InvoiceLine>

<InvoiceLine>
   <LineNumber>3</LineNumber>
<Text>Room Charges</Text>
<Amount>90</Amount>
<TransactionDate>6-DEC-06</TransactionDate>
</InvoiceLine>

<InvoiceLine>
   <LineNumber>4</LineNumber>
<Text>Taxes</Text>
<Amount>10</Amount>
<TransactionDate>6-DEC-06</TransactionDate>
</InvoiceLine>

  </InvoiceLine>

<InvoiceLine>
<LineNumber>4</LineNumber>
<Text>XFer from Charles Smith</Text>
```

```
<Amount>200</Amount>
<TransactionDate>6-DEC-06</TransactionDate>

<InvoiceLine>
   <LineNumber>1</LineNumber>
<Text>Room Charges</Text>
<Amount>90</Amount>
<TransactionDate>5-DEC-06</TransactionDate>
</InvoiceLine>

<InvoiceLine>
   <LineNumber>2</LineNumber>
<Text>Taxes</Text>
<Amount>10</Amount>
<TransactionDate>5-DEC-06</TransactionDate>
</InvoiceLine>

<InvoiceLine>
   <LineNumber>3</LineNumber>
<Text>Room Charges</Text>
<Amount>90</Amount>
<TransactionDate>6-DEC-06</TransactionDate>
</InvoiceLine>

<InvoiceLine>
   <LineNumber>4</LineNumber>
<Text>Taxes</Text>
<Amount>10</Amount>
<TransactionDate>6-DEC-06</TransactionDate>
</InvoiceLine>
  </InvoiceLine>

</PostAR>
```


Resulting Accounts :

Guest Ledger

Debit	Credit
	600 (1)

CityLedger or A/R

Debit	Credit
600 (1)	

Chapter 33 General attributes of the Back Office - Interface

The interface described in this document is intended to become a standard interface which can be used to send basic accounting and statistical data from various hotel specific systems to accounting systems run in the hotel back office.

The interface is based on a set of web services. The messages which are interchanged between the interconnected systems are formatted as SOAP documents, as described in HTNG Framework 2.0 document.

An important characteristic of the messages described in this document is that they modifiy or add data in an accounting system. These are highly critical data, which must not be corrupted by processing a message twice or by loosing a message.

Since Framework 2.0 does not support advanced Web Service protocols like WS Transaction or WS ReliableMessaging, additional measures have to be taken by the communication partners using these messages to ensure that the communication is reliable.
In order to make this task easier for the potential implementors, a 'MessageNumber' element is included in each message header. The element MessageNumber must be kept unique by the sender of a message. If a message has to be repeated, e.g. due to transmission problems, the same MessageNumber has to supplied. If a message is received in the Back Office system with a MessageNumber which has already been received before, the Back Office system has to discard this message and to send a positive acknowledgement in order to prevent further repetitions.

The interface is not designed with a specific posting model in mind. The implementation of a specific posting model is left up to the mapping layer. Examples of postings resulting from messages sent to the accounting system can be found in the separate document "Use Cases".

33.1 Message Flow

Figure 1 shows the message flow between systems in a heterogeneous hotel system landscape.

Messages generated by revenue capturing systems other than a PMS could be handled in two different ways:

A. They could first be sent to the PMS , be consolidated by the PMS and then sent to the mapping layer
B. They could be sent directly to the mapping layer, without being processed by the PMS Evidently a mixture of both approaches would be possible

The messages as defined in this document are supporting alternative B.

Note: This specification is included for information and reference only. It is not included in the Certification Program at this time. It is subject to change in the future.

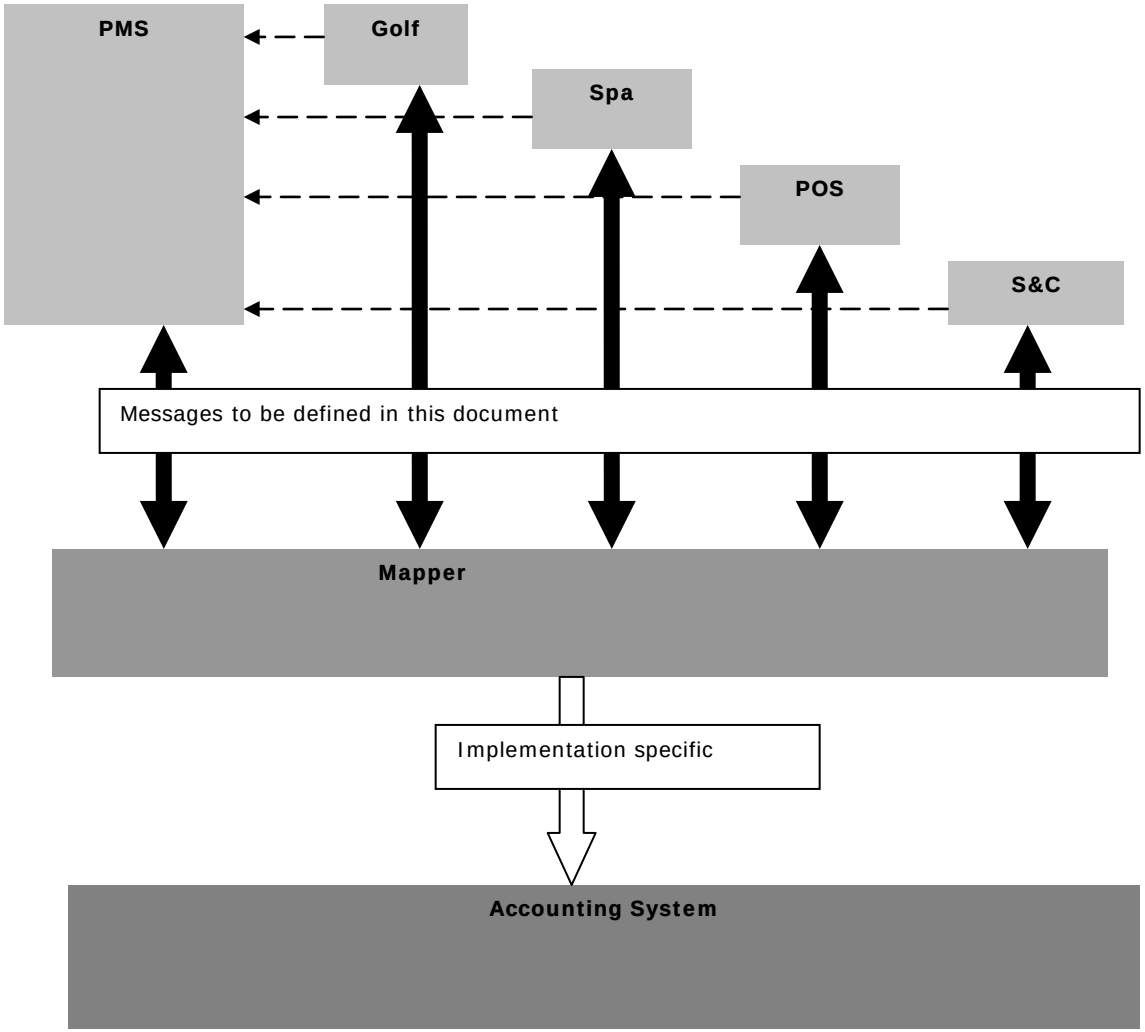


Figure 1: Message Flow Diagram

Chapter 34 Message Overview

34.1 Predefined Dimensions

Dimensions are elements of the PostGL and PostStatistics message. Dimensions are used to transmit additional information related to the transaction codes.
The set of dimensions and the related values is not restricted in this standard.
The following dimensions are understood as proposals for different types of revenue capturing systems:

Retail:

Dimension	Value
RevenueCenter	Hotel specific name of a store
Shift	Time period within a whole day (e.g. am, pm)
CustomerType	e.g. resident, non-resident

F&B:

Dimension	Value
RevenueCenter	Hotel specific name of a restaurant
RevenueCenterType	E.g. Bar, Self Service…
Meal Period	e.g. Breakfast,Lunch, Dinner
CustomerType	e.g. resident, non-resident

Hotel:

Dimension	Value
Market Segment	Luxury, Business…
BookingSource	e.g. eMail, Phone, Walk in, Internet
RoomType	e.g. Single, Double, Suite
CustomerType	e.g. resident, non-resident

Spa:

Dimension	Value
ActivityType	e.g. Massage, Body_Treatment, Salon_Treatment
ProviderType	Internal, External, Therapist qualification
FacilityType	Pool, Sauna, Solarium
EquipmentType	

Golf:

Dimension	Value

CustomerType	e.g. resident, non-resident
Package	
GolfClub	
GolfCourse	
Feetype	e.g. Green fee, member fee….

Entertainment

Dimension	Value
CustomerType	e.g. resident, non-resident
Entertainment_type	e.g. Minigolf, Horse_Riding

34.2 Message Descriptions

34.2.1 PostGL

This message is used to send different kinds of accounting data (revenues, payments, adjustments)

Note that there is no cancel message. In order to reflect changes, e.g. in a folio after night audit, values may be negative or can be posted inverse by using the Debit/Credit Marker field in the message.

Top Level Element	Second level Element	Type	Required	Comment
Header		Complex single occurrence	Yes	This element has an attribute TransactionMode: TransactionMode='SingleSide' specifies, that the Back Office has to use the transaction_code contained in either in the credit- or debit element to build the complete posting record. TransactionMode='PostingRecord' specifies, that the accounts specified in the debit and credit elements are forming a posting record. In that case, only the element TransactionAccount Number should be used.
	MessageNumber	String	Yes	This element is used to prevent that a message is processed twice in the back office system. Each RCS has to supply a unique MessageNumber in every message (must be unique together with SourceSystemId). If the RCS wants to re-send a message due to transmission errors, the same MessageNumber as in the precendent message has to be provided. The Back

				Office system checks if a message with the same MessageNumber has already been processed. If this is the case the message will be discarded and a positive response will be send.
	PropertyCode	String	No	Only needed when multiple company codes have to be supported in accounting
	TransactionDate	Datetime	Yes	To be used as posting date
	ServiceDate	Datetime	No	Date of service
	SourceSystemType	String	Yes	e.g. Hotel, Spa, Golf, Entertainment…
	SourceSystemId	String	Yes	Unique Identifier for revenue capturing system
	User	String	No	User in sending system
	CustomerCode	String	No	Identification of debtor account
	Reservation#	String	No	Reservation number Is unique together with source system id
Transaction		Complex Multiple occurrence	Min 1	
	Credit	Complex Multiple occurrence	No	For allowed subelements see Table 2
	Debit	Complex Multiple occurrence	No	For allowed subelements see Table 2
	Dimension	Complex	No	Subelements are the elements <DimensionName> and

				<DimensionValue> Values of the elements DimensionName and DimensionValue are of data type STRING and not restricted. There are no mandatory values. For proposed dimension elements see Chapter 4.1 “Predefined Dimensions” Example: <Dimension> <DimensionName>MarketSegment</DimensionName> <DimensionValue>Double </DimensionValue> </Dimension>>
		String	No	DimensionName
		String	No	DimensionValue

Table 1

Allowed subelements for debit and credit element

TransactionCode	string	No	Together with the dimensions, this field is used by the mapping system to determine the account number.
TransactionAccount Number	String	No	In case mapping has already been done by sender.
TransactionNumber	Number	No	Transaction in sending system
LineNumber	Number	No	Line of transaction in sending system
Amount	Number of decimals corresponding to currency	Yes	

Currency	Currency	Yes	As defined in ISO standard, eg.USD, EUR
TransactionText	String	Yes	Reference text

Table 2

34.2.2 PostAR

Top Level	Second Level	Type	Required	Comment
Header		Complex		
	MessageNumber	String	Yes	See PostGL
	PropertyCode	String	No	Used if there are multiple properties.
	Date	String	Yes	
	SourceSystemType	String	Yes	e.g. Hotel, Spa, Golf, Entertainment…
	SourceSystemId	String	Yes	Unique Identifier for revenue capturing system
	User	String	No	User in sending system
	CustomerCode	String	No	Is unique together with source system id. For group Invoices, this element can be appear several times in the message
	Reservation#	String	No	
Invoice		Complex		
	InvoiceNumber	String	No	Is not necessary if invoices are assigned a new number in the backoffice system.
	InvoiceDate	Datetime	No	Can default to system date if not specified.
	DueDate	Datetime	No	Can default to payment terms set up for customer in back office system
	TotalInvoice Amount	Number	Yes	This is the amount left to pay for the customer. It's the total invoice amount minus prepayments, intermediate payments etc.
	Currency	String	No	
	Dimension	Complex	No	Subelements are the elements <DimensionName> and <DimensionValue> Values of the elements DimensionName and DimensionValue are of data type STRING and not restricted. There are no mandatory values.

				For proposed dimension elements see Chapter “Predefined Dimensions” Example: <Dimension> <DimensionName>MarketSegment</DimensionName> <DimensionValue>Double </DimensionValue> </Dimension>
	Tax	Complex	No	Subelements are <TaxRate> <TaxBase> <TaxAmount> <TaxType> <TaxIncluded> Example: <Tax> <Taxrate>A1</TaxRate> <TaxBase>45678</TaxBase> <TaxAmount>1234</TaxAmount> <TaxIncluded>TRUE</TaxIncluded> </Tax> <Tax> <Taxrate>A2</TaxRate> <TaxBase>1234</TaxBase> <TaxAmount>12 </TaxAmount> <TaxIncluded>FALSE</TaxIncluded> </Tax> The tax section is repeatable because in some countries (e.g. Belgium) the invoice may have only one invoice amount, but the invoice total may be split up over up to three different tax bases/rates/amounts
	Text	String	No	Comment text
	InvoiceLine	Complex	No	This section can be used if detailed transaction lines are required rather than just the invoice total. Subelements are: <LineNumber>, e.g. 1 <FolioNumber> <Dimension>, e.g. Room Revenue Corporate <Amount>, e.g. $100 <Currency>, e.g. USD <TransactionDate> e.g. 12 July 2007 <ServiceDate> e.g. 13 July 2007 <Text> e.g. room charge night of 12 july <Tax type> e.g. VAT 25% <Tax Rate>, e.g. 25%

				<Tax Base>, e.g. $80 <Tax Amount>, e.g. $20 If this detailed InvoiceLine section is used, then obviously the sum of the line amounts needs to match the invoice total amount. Dimension, currency, date, and tax information specified at the line level overrides data specified at the invoice level. Note: InvoiceLine can be used recursively, i.e. InvoiceLine can be a subelement of InvoiceLine

Table 3

34.2.3 UpdateCustomerData

This message is used to update or create a customer as a debtor in the accounting system.
It contains the customer number as defined by the sender.

Top Level Element	Second level element	Type	Required	Comment
Header		complex		
	MessageNumber	String	Yes	See PostGL
	PropertyCode	String	No	Only needed when multiple company codes have to be supported in accounting
	Date	Datetime	Yes	Posting date
	SourceSystemType	String	Yes	e.g. Hotel, Spa, Golf, Entertainment…
	SourceSystemId	String	Yes	Unique Identifier for revenue capturing system
	User	String	No	User in sending system
	CustomerCode	string	Yes	

CustomerAddress		complex	Yes	See document "Guest Profile Elements"
PersonName		complex	Yes	See document "Guest Profile Elements"
BusinessTitle		string	Yes	See document "Guest Profile Elements"
Phone		complex	No	See document "Guest Profile Elements"
PreferenceList		complex	No	See document "Guest Profile Elements"
InvoiceAddress n		complex	No	
ContactPerson		complex	No	

Table 4

34.2.4 PostStatisics

This message is used to post statistical data not belonging to a financial transaction

Top Level Element	Second Level Element	Type	Required	Comment
Header				
	MessageNumber	String	Yes	See PostGL
	PropertyCode	String	No	Only needed when multiple company codes have to be supported in accounting
	Date	Date time	Yes	Posting date
	SourceSystemType	String	Yes	e.g. Hotel, Spa…
	SourceSystemId	String	Yes	Unique Identifier for revenue capturing system
	User	String	No	User in sending system
Statistic		Complex	Yes min. 1	For proposed StatisticalDataName values see Chapter "Examples for Standard Statistical Data Names" Example: < DataDescription> <StatisticalDataName> Hotel_Number_of_Guests

				</StatisticalDataName> <StatisticalDataValue> 8264 </StatisticalDataValue> </DataDescription
	StatisticalData	Complex	yes	Subelements are <Description> < Value>.
	Dimension	Complex	No	Subelements are <DimensionName> <DimensionValue> For proposed dimension elements see Chapter “Predefined Dimensions” Example: <Dimension> <DimensionName>MarketSegment></DimensionName> <DimensionValue>Double</DimensionValue> </Dimension

Table 5

Examples for Standard Statistical Data Names

Hotel_Number_of_Guests	integer
Hotel_Number_of_Covers	integer
Hotel_Rooms_Availabe	integer
Hotel_Rooms_Out_of_order	integer
Hotel_Rooms_out of_service	integer
Hotel_Rooms_occupied	integer
Hotel_Complementary_rooms	integer
Hotel_House_rooms	integer
Hotel_Number_of_Adults	integer
Hotel_Number_of_Children	integer
Hotel_Number_of_Groups	integer
Hotel_Number_of_Permanent-Guests	integer
Spa_Number_of_Walkins	integer
Spa_Number_of_Hotel_Guests	integer
Spa_Number_of_club_members	integer
Spa_Number_of_residence_guests	integer
Spa_Treatment_Room_Hours_Avalilable	integer
Spa_Treatment_Room_Hours_Used	integer

Spa_Available_Treatment_Rooms	integer

34.2.5 GetMapping

This message can be used by the RCS to ask if the mapping system supports specific combination of transaction code and dimensions.

Top Level Element	Second Level Element	Third Level Element	Type	Required	Comment
Header					
	MessageNumber		String	Yes	See PostGL
	PropertyCode		String	No	Only needed when multiple company codes have to be supported in accounting
	Date		Date time	Yes	Posting date
	SourceSystemType		String	Yes	e.g. Hotel, Spa…
	SourceSystemId		String	Yes	Unique Identifier for revenue capturing system
	User		String	No	User in sending system
Mapping			Complex, multiple	Yes min. 1	
	Code		string	yes	Since there are different mappings for TransactionCodes and StatisticalDataNames, an attribute ‘ ‘MappingType’ is used to differentiate between these mapping sets. The Attribute has two possible values: TransactionCode StatisticalDataName Example: <Code MappingType = ‘TransactionCode’> RoomRev </code>
	Dimension		complex	No	For proposed dimension elements see Chapter 4.1 “Predefined Dimensions” Example: <Dimension> <DimensionName> MarketSegment> </DimensionName>

					<DimensionValue> Double </DimensionValue> </Dimension>
		Dimension Name	string	No	
		Dimension Value.	string	No	
	Validation Result		complex	Min 1	Is provided in the response message
		Credit Account Number	String	No	Example: <validationResult> <DebitAccountNumber> 611111 </ DebitAccountNumber <CreditAccountNumber>114456 </Credit AccountNumber > </validationresult>
		DebitAccountNumber	String	No	Example: <validationResult> <DebitAccountNumber> 611111 </ DebitAccountNumber> <CreditAccountNumber>114456 </ CreditAccountNumber > </validationresult>
		Credit Dimension	complex	No	Subelements are the elements <DimensionName> and <DimensionValue> Example: <validationResult> <DimensionName>ProfitCenter</ProfitCenter> <DimensionValue>4711</DimensionValue> </validationresult>

		Debit Dimension	complex	No	Subelements are the elements <DimensionName> and <DimensionValue> Example: <validationResult> <DimensionName>ProfitCenter</ProfitCenter> <DimensionValue>4711</DimensionValue> </validationresult>

Table 6

Chapter 35 Message Specifications

35.1 Back Office Service

The WSDL for the HTNG Back Office Integrations System defines the "BackOfficeService" web service. This web service consists of five functions defined in one port type. All messages originate from the source revenue capturing system, and are transmitted to the accounting system. The port types are:

- **AccPortType**
  - Functions provided by the accounting system for posting various transactions.

35.1.1 Supporting Schemas

The Back Office Web Service imports three schemas which define the data payloads required by the functional messages. These are:

- **Common**.xsd
  - Defines generic types used by various other elements.
- **Name**.xsd
  - Defines data types specific to a guest profile.
- **Transaction**.xsd
  - Defines general transaction elements.

Common elements used in several messages are included in the Common Schema Elements section of this document.

35.1.2 Soap Header

All messages must include a soap header that conforms to the WSAddressing and WSSecurity specifications. Receiving systems may optionally require that the wsa:To element match a specific destination address, and therefore, this element should be configurable at run time. In addition, all sending systems must identify themselves by specifying a wsa:From element unique for their system. Typically, this is in the form of "URN:*<system>*". The wsa:ReplyTo address may be the anonymous form described in the August, 2004 specification.

WSSecurity may be optionally implemented. Providers are only required to support basic plain text authentication with a username and password. The user credentials will be limited to a single login per system, agreed between two vendors. There is no requirement to support multiple user logins from a single vendor through this interface.

35.1.2.1 Sample Header Message

```
<soap:Header>
   <wsa:Action>http://htng.org/PWSWG/2006/08/SingleGuestItinerary#PostPayment</wsa:Action>
   <wsa:From>
      <wsa:Address>urn:SPASOFT</wsa:Address>
   </wsa:From>
   <wsa:MessageID>urn:uuid:e63d962e-94b6-434c-89ea-1c5ae2e0f8ba</wsa:MessageID>
   <wsa:ReplyTo>
      <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
   </wsa:ReplyTo>
   <wsa:To>http://www.micros.com/HTNGActivity/</wsa:To>
```

```
   <wsse:Security soap:mustUnderstand="1">
      <wsu:Timestamp wsu:Id="Timestamp-015e2941-114a-466e-87b6-8c431b9f5c44">
         <wsu:Created>2006-10-26T12:49:17Z</wsu:Created>
         <wsu:Expires>2006-10-26T12:54:17Z</wsu:Expires>
      </wsu:Timestamp>
      <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="SecurityToken-627f1ab1-338a-451f-9829-84f248e57ad8">
         <wsse:Username>HTNG</wsse:Username>
         <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">HTNG</wsse:Password>
         <wsse:Nonce>qls5nr9rM7VaUYAwhrHSoA==</wsse:Nonce>
         <wsu:Created>2006-10-26T12:49:17Z</wsu:Created>
      </wsse:UsernameToken>
   </wsse:Security>
</soap:Header>
```

35.1.3 Asynchronous and Asynchronous Processing

The specification is originally designed to implement synchronous communication. In the future, the system may be extended to implement asynchronous processing, per standards established in the HTNG Framework 2.0 specification.

35.1.4 Mapping System

Inherit in any back office integration is the requirement of a mapping system which translates accounting (transaction) codes used in the source system to those of the accounting system. The mapping system may be provided as a layer of either the source system (e.g. the property management system) or by the accounting system. As an alternative, the mapping may be performed by a separate system. In any instance, it should act as a pass-through mechanism, adding account detail to the message.

35.1.5 Roles

There are various roles that interact with this interface. These are:

- Accounting System (ACC) – collects summary or detailed information from various source systems.
- Mapping Layer (MAP) – translates account identifiers in the source system to equivalent items in the accounting system.
- Revenue Capturing System (RCS) – generic system generating source transactions. Examples include:
    - Property Management System (PMS) – generates source transactions from a hotel property which are passed to the accounting system either in detail form or as summary information.
    - Point of Sale System (POS) – generates source transactions from a point of sale collection system which are passed to the accounting system.
    - Activity (ACT) – generates source transactions from entities such as a spa or golf booking system.

35.2 Acc Port Type

The Acc port is provided by the accounting system in order to provide a means to post various transactions generated in a source system. The functions provided are:

- PostGeneralLedger
    - Post generic transaction to the accounting system.

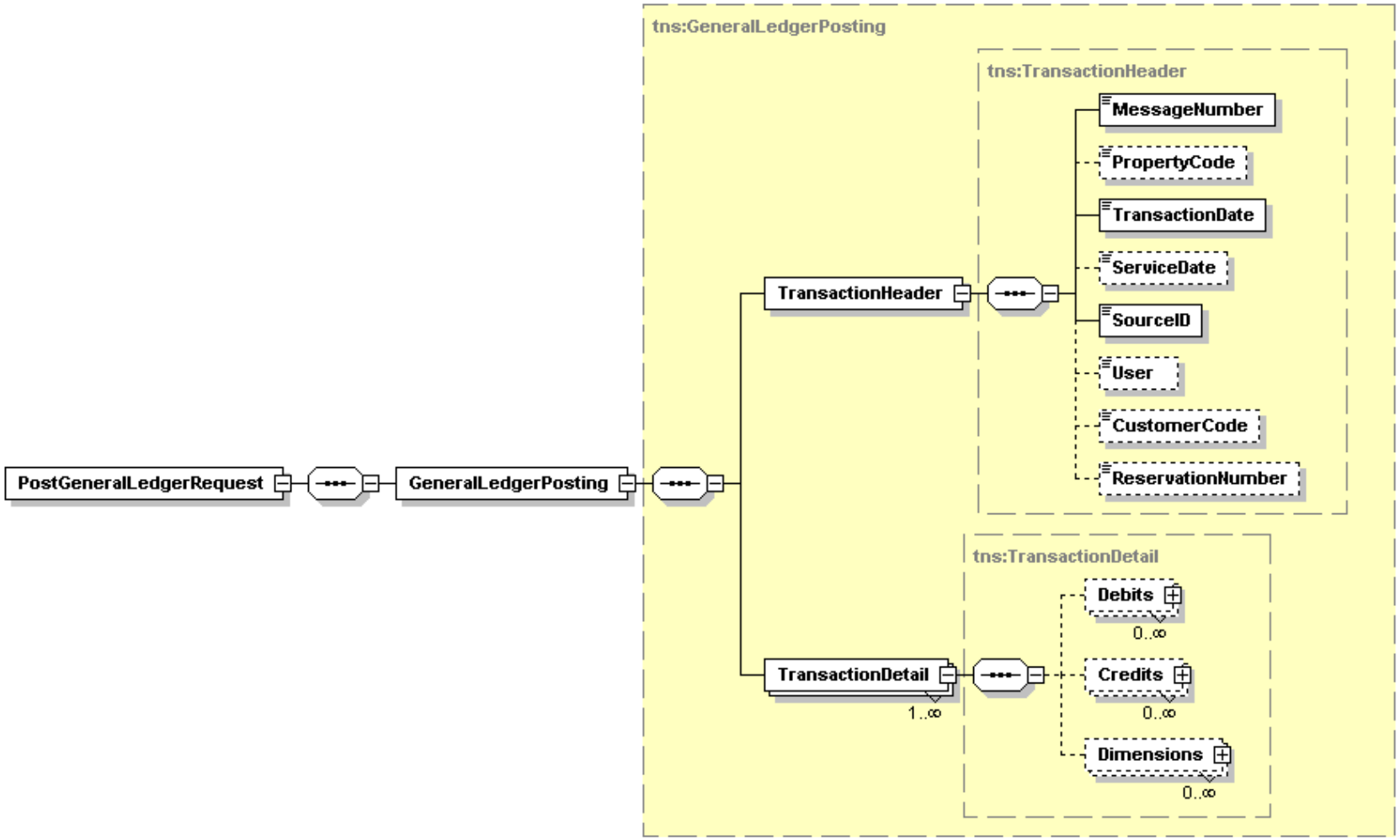
- PostAccountsReceivable
  - Post invoices to the accounting system.
- UpdateCustomer
  - Send customer information to the accounting system.
- PostStatistics
  - Send statistical information to the accounting system.
- GetMapping
  - Request validation of mapping from the accounting system.

35.2.1 Post General Ledger

The Post General Ledger function sends transactional information from the source revenue capturing system to the accounting system. This may be done on an individual transaction basis, or as summary postings. For instance, a hotel might post all of its revenue and payments as a daily summary created during the night audit process. The result returned from this message is a ResultStatus response as defined in the Common Data Elements section.

Port	AccPortType
Binding	AccBinding
Operation	PostGeneralLedger
Soap Action	http://htng.org/PWSWG/2007/09/BackOfficeIntegration#PostGeneralLedger
Input	PostGeneralLedgerRequest
Output	PostGeneralLedgerResponse
Primary Schema	Transactoin.xsd

35.2.1.1 PostGeneralLedgerRequest



PostGeneralLedgerRequest

```
<xs:element name="PostGeneralLedgerRequest">
    <xs:complexType>
```

```
        <xs:sequence>
            <xs:element name="GeneralLedgerPosting" xmlns:q1="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
                type="q1:GeneralLedgerPosting" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

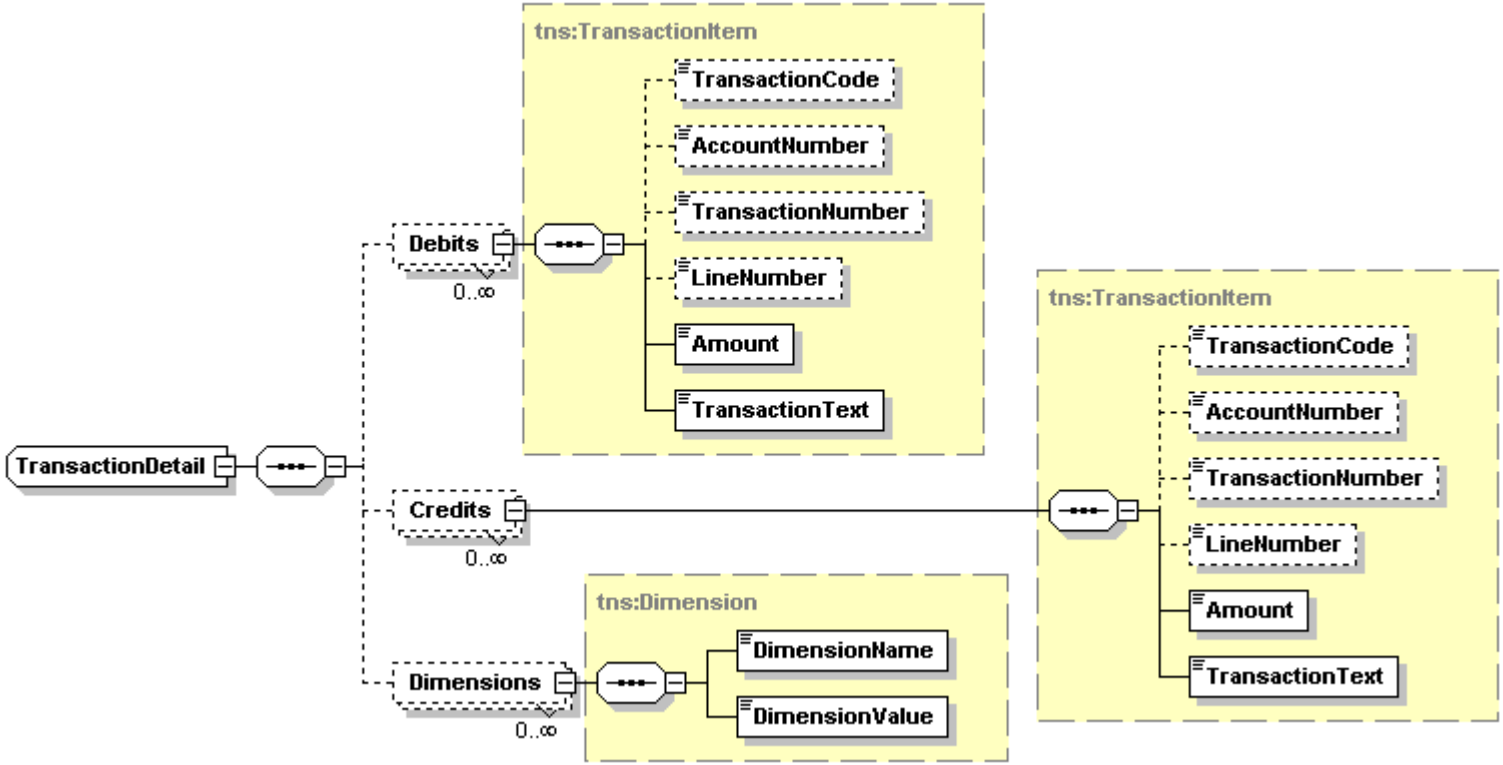
Name	Type	Data Type	Use	Comments
GeneralLedgerPosting	element	GeneralLedgerPosting	required	General ledger posting transaction

GeneralLedgerPosting

GeneralLedgerPosting

```
<xs:complexType name="GeneralLedgerPosting">
    <xs:sequence>
        <xs:element name="TransactionHeader" xmlns:q1="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
            type="q1:TransactionHeader" />
        <xs:element maxOccurs="unbounded" name="TransactionDetail"
            xmlns:q2="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types" type="q2:TransactionDetail" />
    </xs:sequence>
    <xs:attribute name="TransactionMode" xmlns:q3="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
        type="q3:TransactionMode" use="required" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
TransactionMode	attribute	TransactionMode	required	Either **SingleSide** or **PostingRecord** to indicate the type of posting.
TransactionHeader	element	TransactionHeader	required	Transaction header (see Common Data Elements)
TransactionDetail	element	TransactionDetail	required / multiple	Transaction detail.



TransactionDetail

TransactionDetail

```
<xs:complexType name="TransactionDetail">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Debits"
            xmlns:q1="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types" type="q1:TransactionItem" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Credits"
            xmlns:q2="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types" type="q2:TransactionItem" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Dimensions"
            xmlns:q3="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types" type="q3:Dimension" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Debits	element	TransactionItem	optional /	Debit(s) for the transaction.

			multiple	
Credits	element	TransactionItem	optional / multiple	Credit(s) for the transaction.
Dimensions	element	Dimension	optional / multiple	Additional dimensions related to the transaction. The accounting system may use this information in conjunction with the transaction code in order to determine the appropriated account to post to.

TransactionItem

TransactionItem

```
<xs:complexType name="TransactionItem">
    <xs:sequence>
        <xs:element minOccurs="0" name="TransactionCode" type="xs:string" />
        <xs:element minOccurs="0" name="AccountNumber" type="xs:string" />
        <xs:element minOccurs="0" name="TransactionNumber" type="xs:string" />
        <xs:element minOccurs="0" name="LineNumber" type="xs:string" />
        <xs:element name="Amount" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:Amount" />
        <xs:element name="TransactionText" type="xs:string" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
TransactionCode	element	string	optional	Transaction code. The accounting system may combine dimensional data to determine the appropriate account
AccountNumber	element	string	optional	Account number. Either the transaction code or the account number must be present.
TransactionNumber	element	string	optional	Transaction number.
LineNumber	element	string	optional	Line number.
Amount	element	Amount	required	Amount of the transaction.
TransactionText	element	string	required	Descriptive text for the transaction.

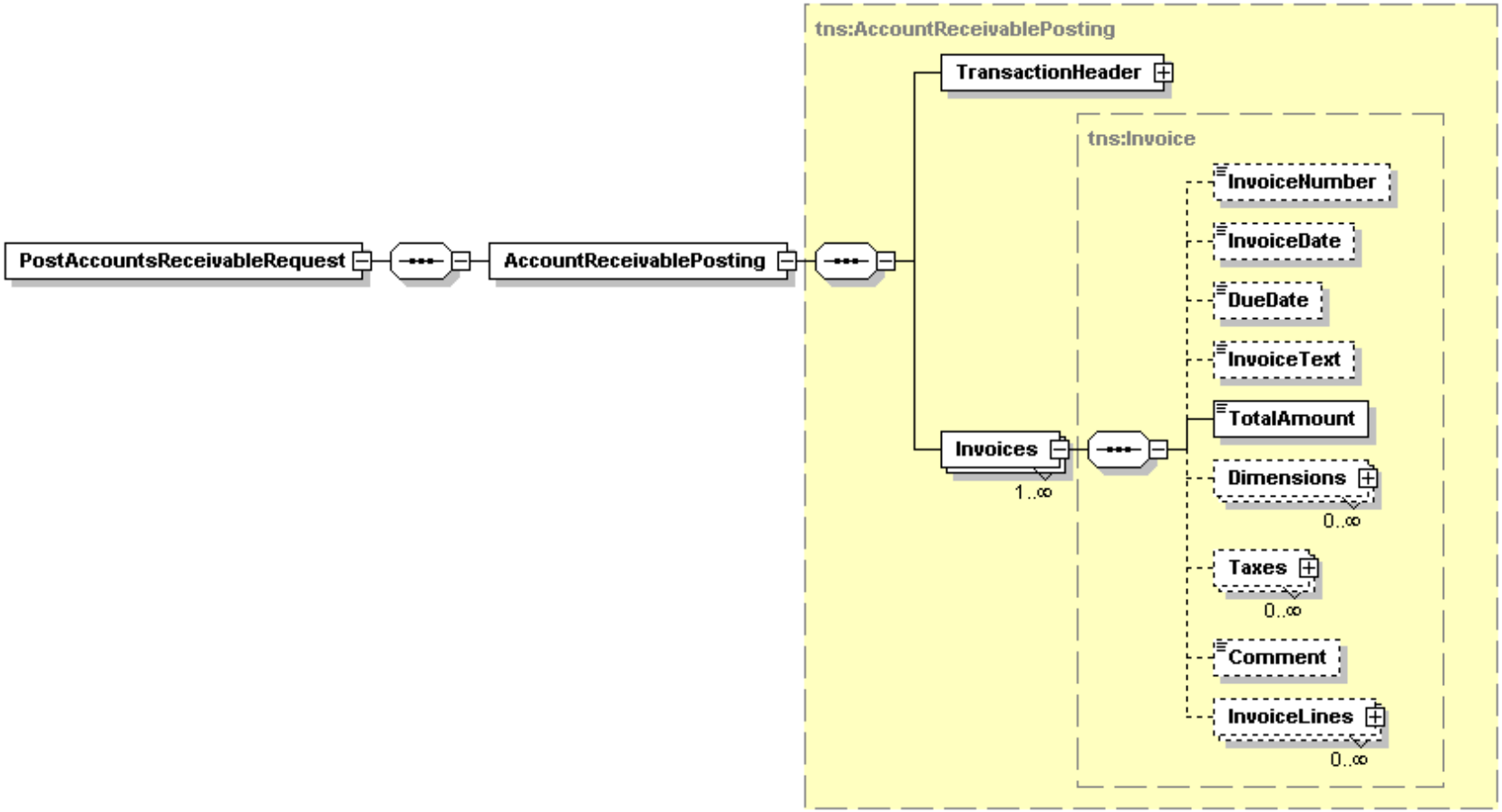
35.2.2 Post Accounts Receivable

The Post Accounts Receivable function is used to send invoice details to the accounting system.

Port	AccPortType
Binding	AccBinding
Operation	PostAccountsReceivable
Soap Action	http://htng.org/PWSWG/2007/09/BackOfficeIntegration#PostAccountsReceivable
Input	PostAccountsReceivableRequest
Output	PostAccountsReceivableResponse

Primary Schema	Transactoin.xsd

35.2.2.1 PostAccountsReceivableRequest



PostAccountsReceivableRequest

```
<xs:element name="PostAccountsReceivableRequest">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="AccountReceivablePosting" xmlns:q1="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
```

```
            type="q1:AccountReceivablePosting" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
AccountReceivablePosting	element	AccountReceivablePosting	required	The account receivable transaction record.

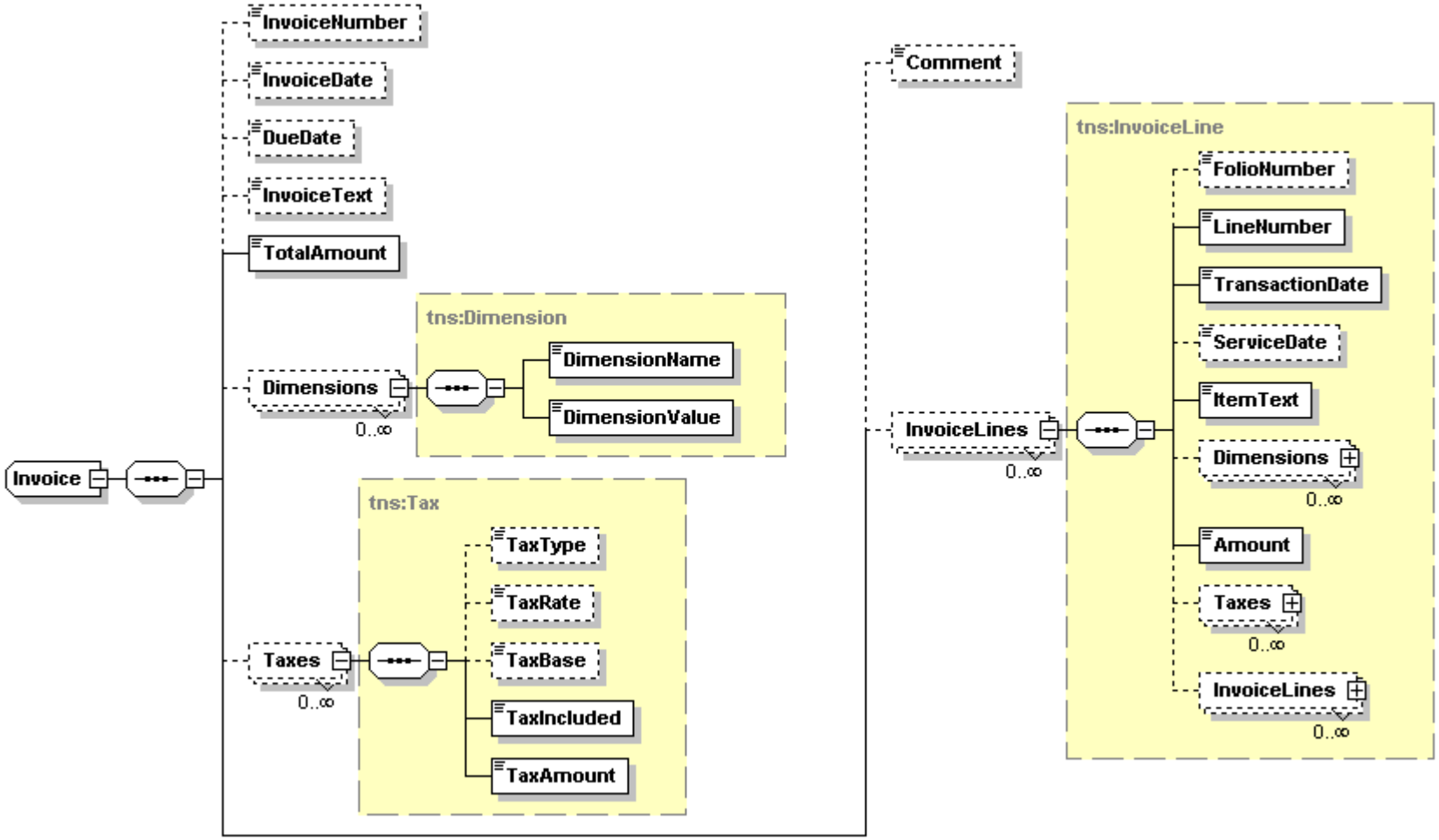
AccountReceivablePosting

AccountReceivablePosting

```
<xs:complexType name="AccountReceivablePosting">
    <xs:sequence>
        <xs:element name="TransactionHeader" xmlns:q1="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
            type="q1:TransactionHeader" />
        <xs:element maxOccurs="unbounded" name="Invoices" xmlns:q2="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
            type="q2:Invoice" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
TransactionHeader	element	TransactionHeader	required	Transaction header (see Common Data Elements)
Invoices	element	Invoice	required / multiple	One or more invoice records.

Invoice



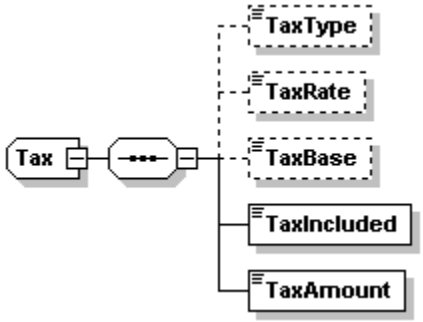
Invoice

`<xs:complexType name="Invoice">`
`  <xs:sequence>`
`  <xs:element minOccurs="0" name="InvoiceNumber" type="xs:string" />`

```
        <xs:element minOccurs="0" name="InvoiceDate" type="xs:dateTime" />
        <xs:element minOccurs="0" name="DueDate" type="xs:dateTime" />
        <xs:element minOccurs="0" name="InvoiceText" type="xs:string" />
        <xs:element name="TotalAmount" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:Amount" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Dimensions"
            xmlns:q2="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types" type="q2:Dimension" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Taxes"
            xmlns:q3="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types" type="q3:Tax" />
        <xs:element minOccurs="0" name="Comment" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="InvoiceLines"
            xmlns:q4="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types" type="q4:InvoiceLine" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
InvoiceNumber	element	string	optional	The invoice number.
InvoiceDate	element	dateTime	optional	The invoice date.
DueDate	element	dateTime	optional	The due date.
InvoiceText	element	string	optional	Invoice text.
TotalAmount	element	Amount	required	The total invoice amount.
Dimensions	element	Dimension	optional / multiple	Additional statistical information.
Taxes	element	Tax	optional / multiple	Optional tax records associated with the invoice.
Comment	element	string	optional	Additional comments.
InvoiceLines	element	InvoiceLine	optional / multiple	Invoice detail records. These are not required if only invoice totals are posted.

Tax

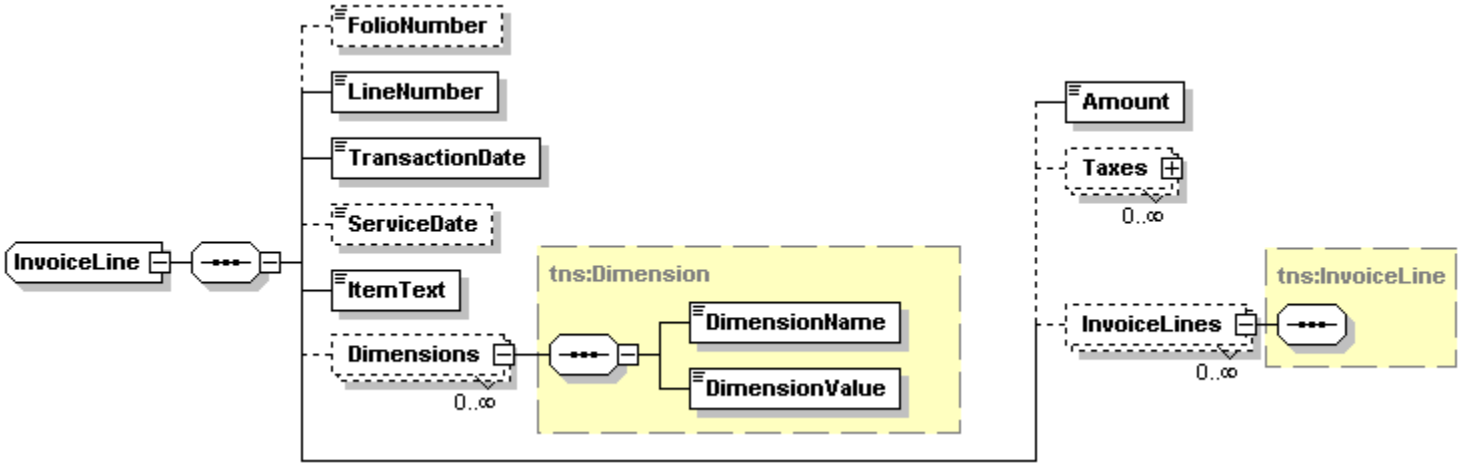


Tax

```
<xs:complexType name="Tax">
    <xs:sequence>
        <xs:element minOccurs="0" name="TaxType" type="xs:string" />
        <xs:element minOccurs="0" name="TaxRate" type="xs:string" />
        <xs:element minOccurs="0" name="TaxBase" type="xs:string" />
        <xs:element name="TaxIncluded" type="xs:boolean" />
        <xs:element name="TaxAmount" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:Amount" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
TaxType	element	string	optional	Tax type.
TaxRate	element	string	optional	Tax rate.
TaxBase	element	string	optional	Tax base amount.
TaxIncluded	element	boolean	required	Indicator whether tax is included in amount or in addition.
TaxAmount	element	Amount	required	Tax amount.

InvoiceLine



InvoiceLine

```
<xs:complexType name="InvoiceLine">
    <xs:sequence>
        <xs:element minOccurs="0" name="FolioNumber" type="xs:string" />
        <xs:element name="LineNumber" type="xs:int" />
        <xs:element name="TransactionDate" type="xs:dateTime" />
        <xs:element minOccurs="0" name="ServiceDate" type="xs:dateTime" />
        <xs:element name="ItemText" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Dimensions"
            xmlns:q1="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types" type="q1:Dimension" />
        <xs:element name="Amount" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q2:Amount" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Taxes"
            xmlns:q3="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types" type="q3:Tax" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="InvoiceLines"
            xmlns:q4="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types" type="q4:InvoiceLine" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
FolioNumber	element	string	optional	Folio number.
LineNumber	element	int	required	Line number.
TransactionDate	element	dateTime	required	Transaction date.
ServiceDate	element	dateTime	optional	Actual service date.
ItemText	element	string	required	Item text.

Dimensions	element	Dimension	optional / multiple	Additional dimensional data.
Amount	element	Amount	required	Item amount.
Taxes	element	Tax	optional / multiple	Tax details.
InvoiceLines	element	InvoiceLine	optional / multiple	Recursive invoice line items.

35.2.3 Update Customer

The Update Customer function allows a revenue capturing system to notify the accounting system when a change occurs in a customer record.

Port	AccPortType
Binding	AccBinding
Operation	UpdateCustomer
Soap Action	http://htng.org/PWSWG/2007/09/BackOfficeIntegration#UpdateCustomer
Input	UpdateCustomerRequest
Output	UpdateCustomerResponse
Primary Schema	Transactoin.xsd

35.2.3.1 UpdateCustomerRequest

UpdateCustomerRequest

```
<xs:element name="UpdateCustomerRequest">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="CustomerUpdate" xmlns:q1="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
                type="q1:CustomerUpdate" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
CustomerUpdate	element	CustomerUpdate	required	Customer update information.

CustomerUpdate

CustomerUpdate

```
<xs:complexType name="CustomerUpdate">
    <xs:sequence>
        <xs:element name="TransactionHeader" xmlns:q1="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
            type="q1:TransactionHeader" />
        <xs:element maxOccurs="unbounded" name="Profile" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
            type="q2:Profile" />
    </xs:sequence>
</xs:complexType>
```

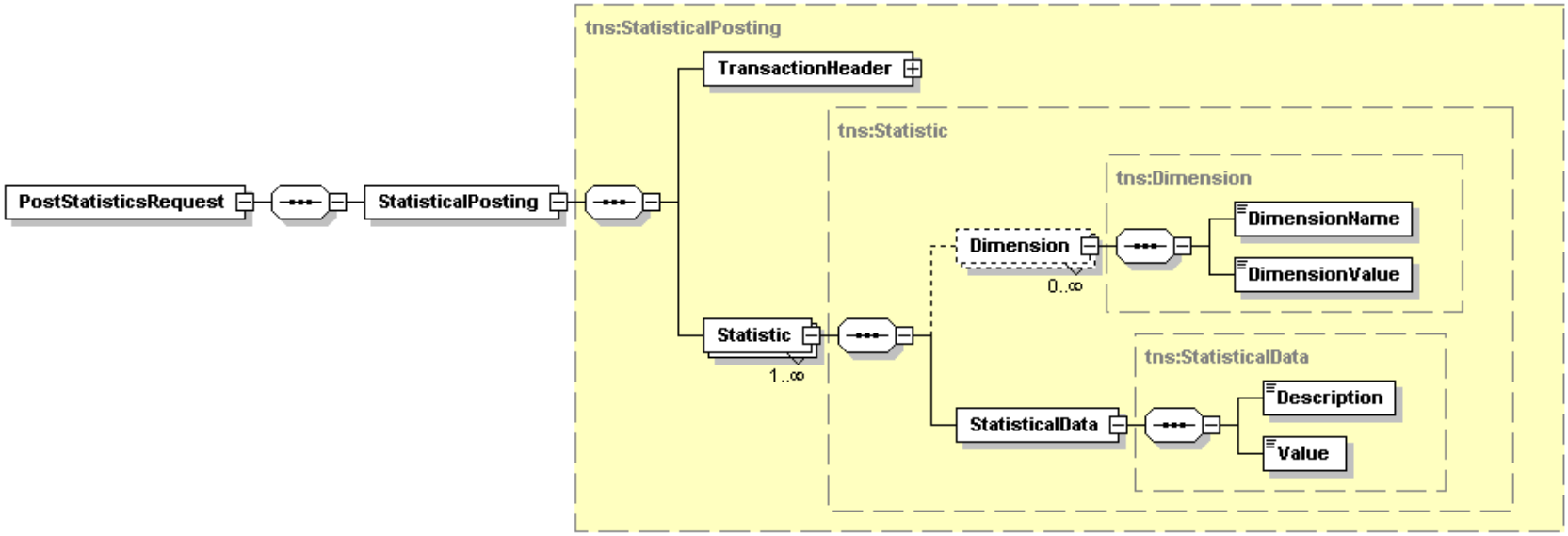
Name	Type	Data Type	Use	Comments
TransactionHeader	element	TransactionHeader	required	Transaction header (see Common Data Elements)
Profile	element	Profile	required / multiple	Customer profile data (see Common Data Elements).

35.2.4 Post Statistics

A revenue capturing system may post various statistical information to the accounting system.

Port	AccPortType
Binding	AccBinding
Operation	PostStatistics
Soap Action	http://htng.org/PWSWG/2007/09/BackOfficeIntegration#PostStatistics
Input	PostStatisticsRequest
Output	PostStatisticsResponse
Primary Schema	Transactoin.xsd

35.2.4.1 PostStatisticsRequest



PostStatisticsRequest				
<xs:element name="PostStatisticsRequest">
<xs:complexType>
<xs:sequence>
<xs:element name="StatisticalPosting" xmlns:q1="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
type="q1:StatisticalPosting" />
</xs:sequence>
</xs:complexType>
</xs:element>				
Name	**Type**	**Data Type**	**Use**	**Comments**
StatisticalPosting	element	StatisticalPosting	required	Statistic record.

StatisticalPosting

StatisticalPosting
<xs:complexType name="StatisticalPosting">
<xs:sequence>

```
        <xs:element name="TransactionHeader" xmlns:q1="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
            type="q1:TransactionHeader" />
        <xs:element maxOccurs="unbounded" name="Statistic" xmlns:q2="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
            type="q2:Statistic" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
TransactionHeader	element	TransactionHeader	required	Transaction header (see Common Data Elements)
Statistic	element	Statistic	required / multiple	Collection of Statistic records

Statistic

Statistic

```
<xs:complexType name="Statistic">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Dimension"
            xmlns:q1="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types" type="q1:Dimension" />
        <xs:element name="StatisticalData" xmlns:q2="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types" type="q2:StatisticalData"
            />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Dimension	element	Dimension	optional / multiple	Dimension records applied to the statistic.
StatisticalData	element	StatisticalData	required	Statistical data record.

StatisticalData

StatisticalData

```
<xs:complexType name="StatisticalData">
    <xs:sequence>
        <xs:element name="Description" type="xs:string" />
        <xs:element name="Value" type="xs:string" />
    </xs:sequence>
</xs:complexType>
```

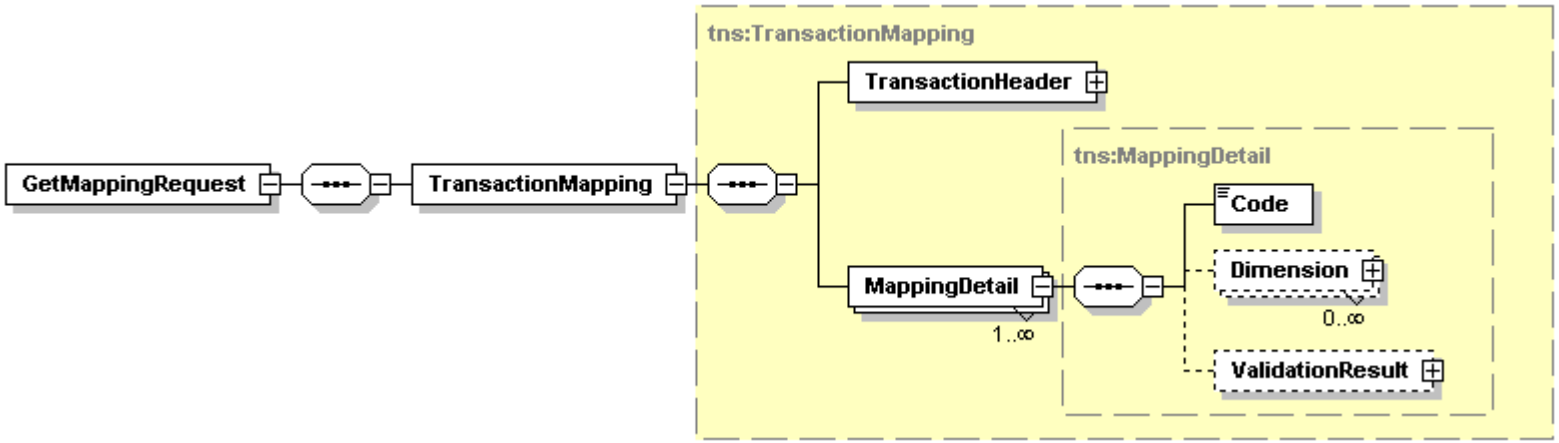
Name	Type	Data Type	Use	Comments
Description	element	string	required	Description of the statistical value.
Value	element	string	required	Statistical value.

35.2.5 Get Mapping

The mapping system needs to link transaction codes used in the revenue capturing system to account codes used in the accounting system. The RCS sends a Get Mapping message to list all of its transaction codes in order to validate that they are all mapped in the accounting system. The accounting systems returns the completed map in the response. This is most likely implemented as an iterative process until all transaction codes are completely mapped.

Port	AccPortType
Binding	AccBinding
Operation	GetMapping
Soap Action	http://htng.org/PWSWG/2007/09/BackOfficeIntegration#GetMapping
Input	GetMappingRequest
Output	GetMappingResponse
Primary Schema	Transactoin.xsd

35.2.5.1 GetMappingRequest



GetMappingRequest

```
<xs:element name="GetMappingRequest">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="TransactionMapping" xmlns:q1="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
                type="q1:TransactionMapping" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments

TransactionMapping	element	TransactionMapping	required	Transaction code mapping

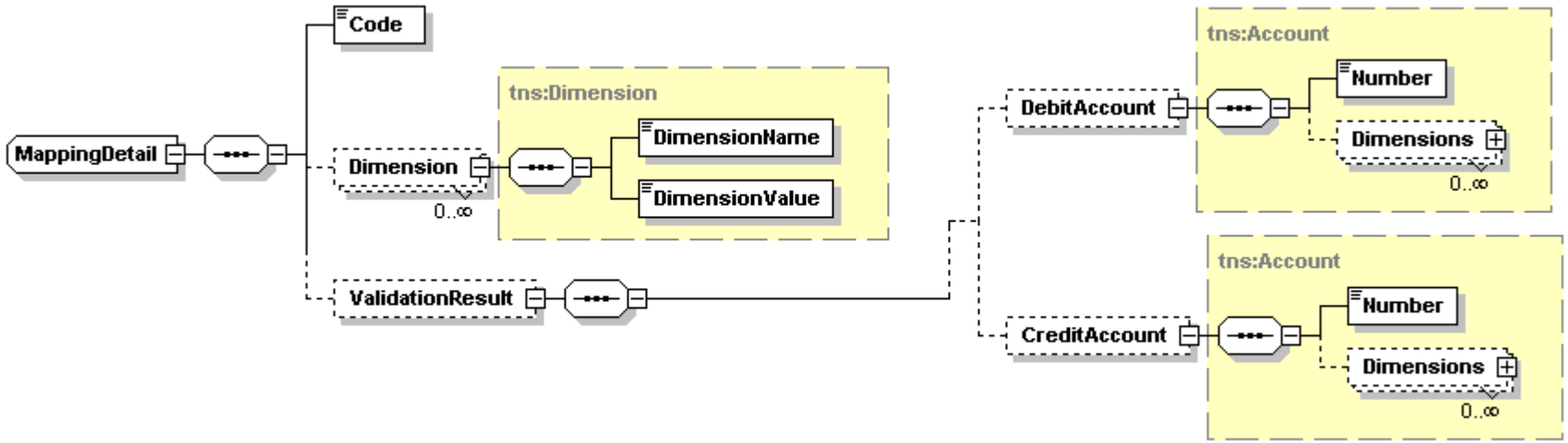
TransactionMapping

TransactionMapping

```
<xs:complexType name="TransactionMapping">
    <xs:sequence>
        <xs:element name="TransactionHeader" xmlns:q1="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
            type="q1:TransactionHeader" />
        <xs:element maxOccurs="unbounded" name="MappingDetail" xmlns:q2="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
            type="q2:MappingDetail" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
TransactionHeader	element	TransactionHeader	required	Transaction header (see Common Data Elements)
MappingDetail	element	MappingDetail	required / multiple	Mapping details.

MappingDetail



MappingDetail

```
<xs:complexType name="MappingDetail">
    <xs:sequence>
        <xs:element name="Code">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="xs:string">
                        <xs:attribute name="mappingType" />
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Dimension"
            xmlns:q1="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types" type="q1:Dimension" />
        <xs:element minOccurs="0" name="ValidationResult">
            <xs:complexType>
                <xs:sequence>
                    <xs:element minOccurs="0" name="DebitAccount" xmlns:q2="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
                        type="q2:Account" />
                    <xs:element minOccurs="0" name="CreditAccount" xmlns:q3="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
                        type="q3:Account" />
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="mapped" type="xs:boolean" use="optional" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
mapped	attribute	boolean	optional	Flag to indicate if transaction code is mapped.
Code	element		required	Transaction or other code in revenue capturing system. Includes attribute (“mappingType”) which defines type of code. “TransactionCode” is one valid type.
Dimension	element	Dimension	optional / multiple	Dimensional details.
ValidationResult	element		optional	Accounting system mapped account numbers.

Account

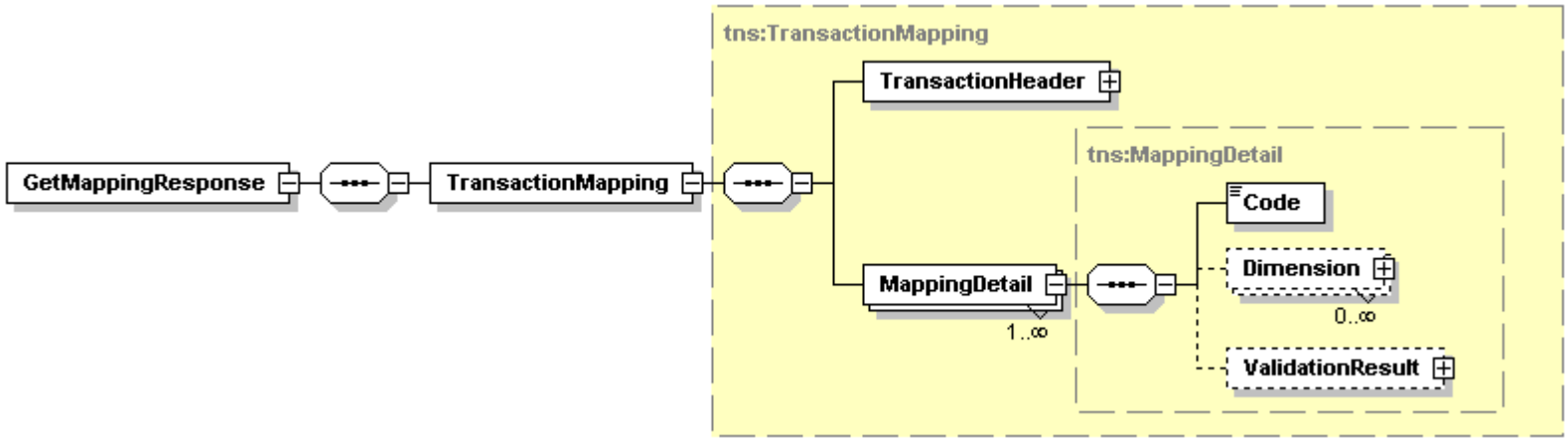
Account

```
<xs:complexType name="Account">
    <xs:sequence>
        <xs:element name="Number" type="xs:string" />
```

```
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Dimensions"
            xmlns:q1="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types" type="q1:Dimension" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Number	element	string	required	Account number.
Dimensions	element	Dimension	optional / multiple	Account dimensions.

35.2.5.2 GetMappingResponse



GetMappingResponse

```
<xs:element name="GetMappingResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="TransactionMapping" xmlns:q1="http://htng.org/PWSWG/2007/09/BackOfficeIntegration/Transaction/Types"
                type="q1:TransactionMapping" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
TransactionMapping	element	TransactionMapping	required	Resultant map.

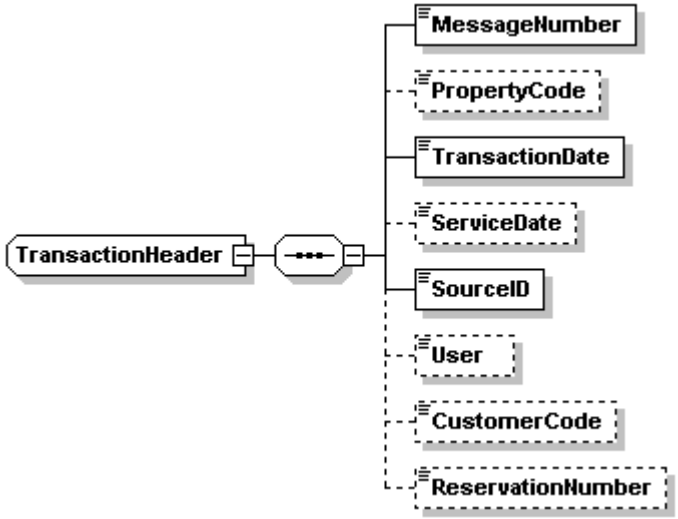
35.3 Common Data Elements

The Back Office Integration Web Service messages share several common data elements. These are consolidated here for reference.

35.3.1 Transaction Schema (Transaction.xsd)

35.3.1.1 TransactionHeader

The Transaction Header is common to all messages posted to the accounting system. It identifies the source of the posting (which may be beyond the message source included in the WSAddressing header), the date of the transaction, and additional relevant information.



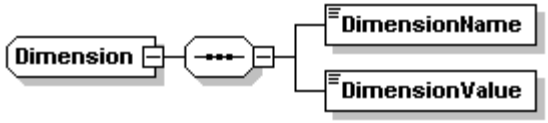
TransactionHeader

```
<xs:complexType name="TransactionHeader">
    <xs:sequence>
        <xs:element name="MessageNumber" type="xs:string" />
        <xs:element minOccurs="0" name="PropertyCode" type="xs:string" />
        <xs:element name="TransactionDate" type="xs:dateTime" />
        <xs:element minOccurs="0" name="ServiceDate" type="xs:dateTime" />
        <xs:element name="SourceID" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueID" />
        <xs:element minOccurs="0" name="User" type="xs:string" />
        <xs:element minOccurs="0" name="CustomerCode" type="xs:string" />
        <xs:element minOccurs="0" name="ReservationNumber" type="xs:string" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
MessageNumber	element	string	required	Unique message number for transaction.
PropertyCode	element	string	optional	Property code.
TransactionDate	element	dateTime	required	Transaction date.

ServiceDate	element	dateTime	optional	Actual service date.
SourceID	element	UniqueID	required	Source identifier.
User	element	string	optional	User code.
CustomerCode	element	string	optional	Customer code.
ReservationNumber	element	string	optional	Reservation number.

35.3.1.2 Dimension



Dimension

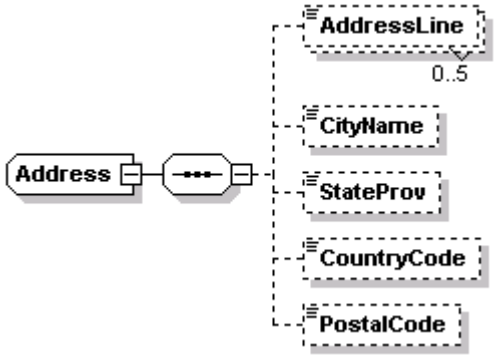
```
<xs:complexType name="Dimension">
    <xs:sequence>
        <xs:element name="DimensionName" type="xs:string" />
        <xs:element name="DimensionValue" type="xs:string" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
DimensionName	element	string	required	
DimensionValue	element	string	required	

35.3.2 Common Schema Elements (Common.xsd)

Namespace	http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types

35.3.2.1 Address



Address

```
<xs:complexType name="Address">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="5" name="AddressLine" type="xs:string" />
        <xs:element minOccurs="0" name="CityName" type="xs:string" />
        <xs:element minOccurs="0" name="StateProv" type="xs:string" />
        <xs:element minOccurs="0" name="CountryCode" type="xs:string" />
        <xs:element minOccurs="0" name="PostalCode" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="addressType" type="xs:string" />
    <xs:attribute name="otherAddressType" type="xs:string" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
addressType	attribute	string	none	Type of address. Typical values are HOME or BUSINESS, but any internally defined value may be used.
otherAddressType	attribute	string	none	Not currently used.
AddressLine	element	string	optional / multiple	Up to five address lines may be specified.
CityName	element	string	optional	City.
StateProv	element	string	optional	State.
CountryCode	element	string	optional	Country.
PostalCode	element	string	optional	Postal code.

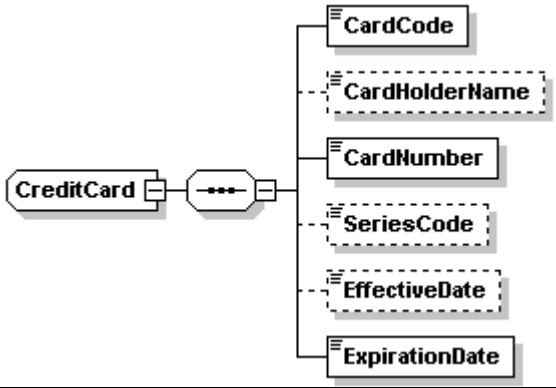
35.3.2.2 Amount

Amount

```
<xs:complexType name="Amount">
    <xs:simpleContent>
        <xs:extension base="xs:double">
            <xs:attribute name="currencyCode" type="xs:string" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Amount	extension	double		The denominational amount.
currencyCode	attribute	string	none	The currency code the amount is expressed in.

35.3.2.3 Credit Card

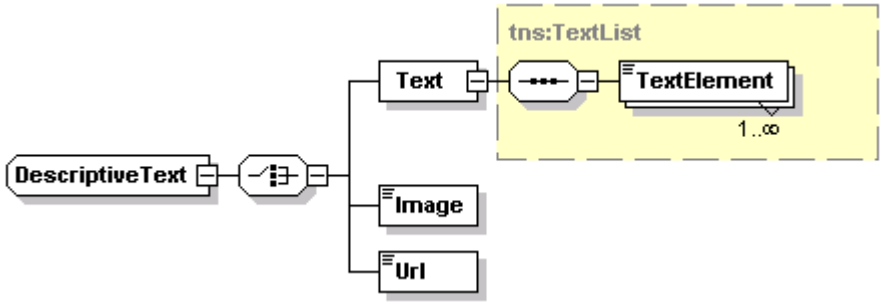


CreditCard

```
<xs:complexType name="CreditCard">
    <xs:sequence>
        <xs:element name="CardCode" type="xs:string" />
        <xs:element minOccurs="0" name="CardHolderName" type="xs:string" />
        <xs:element name="CardNumber" type="xs:string" />
        <xs:element minOccurs="0" name="SeriesCode" type="xs:string" />
        <xs:element minOccurs="0" name="EffectiveDate" type="xs:date" />
        <xs:element name="ExpirationDate" type="xs:date" />
    </xs:sequence>
    <xs:attribute name="cardType" type="xs:string" />
    <xs:attribute name="otherCardType" type="xs:string" />
```

</xs:complexType>				
Name	**Type**	**Data Type**	**Use**	**Comments**
cardType	attribute	string	none	Credit card type.
otherCardType	attribute	string	none	Not used.
CardCode	element	string	required	Credit card code.
CardHolderName	element	string	optional	Card holder name.
CardNumber	element	string	required	Credit card number.
SeriesCode	element	string	optional	Not used.
EffectiveDate	element	date	optional	Effective date.
ExpirationDate	element	date	required	Expiration date.

35.3.2.4 Descriptive Text

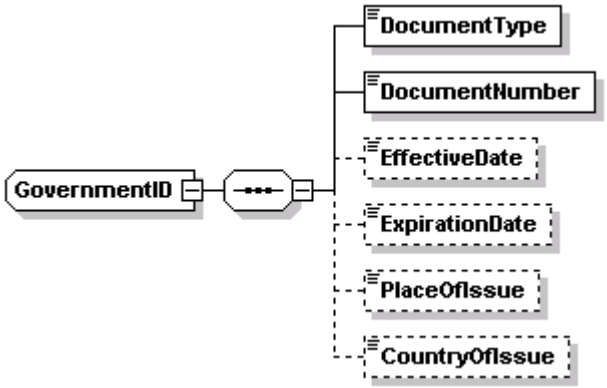


DescriptiveText

```
<xs:complexType name="DescriptiveText">
    <xs:choice>
        <xs:element name="Text" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:TextList" />
        <xs:element name="Image" type="xs:string" />
        <xs:element name="Url" type="xs:anyURI" />
    </xs:choice>
</xs:complexType>
```

Name	**Type**	**Data Type**	**Use**	**Comments**
Text	element	TextList	required	Collection of Text elements (Choice)
Image	element	string	required	Image value (Choice).
Url	element	anyURI	required	URL (Choice).

35.3.2.5 GovernmentID

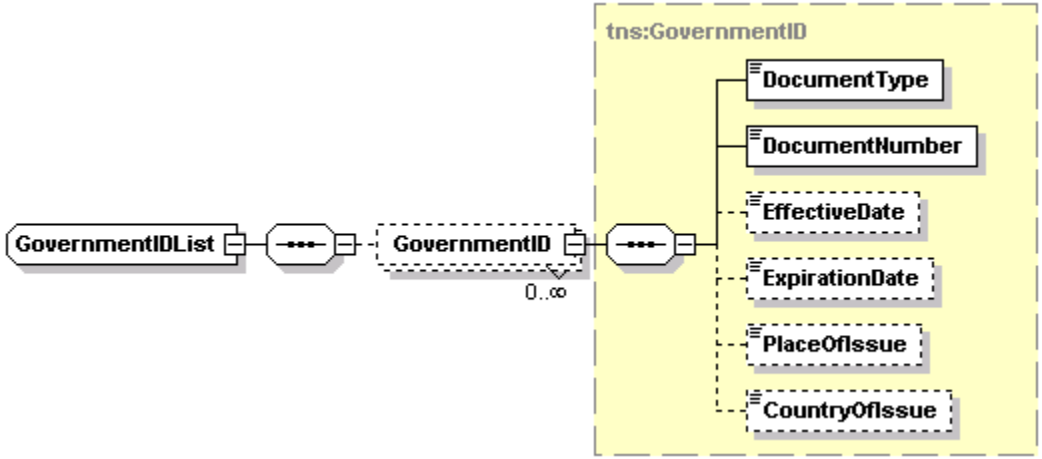


GovernmentID

```
<xs:complexType name="GovernmentID">
    <xs:sequence>
        <xs:element name="DocumentType" type="xs:string" />
        <xs:element name="DocumentNumber" type="xs:string" />
        <xs:element minOccurs="0" name="EffectiveDate" type="xs:date" />
        <xs:element minOccurs="0" name="ExpirationDate" type="xs:date" />
        <xs:element minOccurs="0" name="PlaceOfIssue" type="xs:string" />
        <xs:element minOccurs="0" name="CountryOfIssue" type="xs:string" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
DocumentType	element	string	required	The type of document (e.g. PASSPORT).
DocumentNumber	element	string	required	The document number.
EffectiveDate	element	date	optional	The effective date of the document.
ExpirationDate	element	date	optional	The expiration date of the document.
PlaceOfIssue	element	string	optional	The place of issue.
CountryOfIssue	element	string	optional	The country of issue.

35.3.2.6 GovernmentIDList

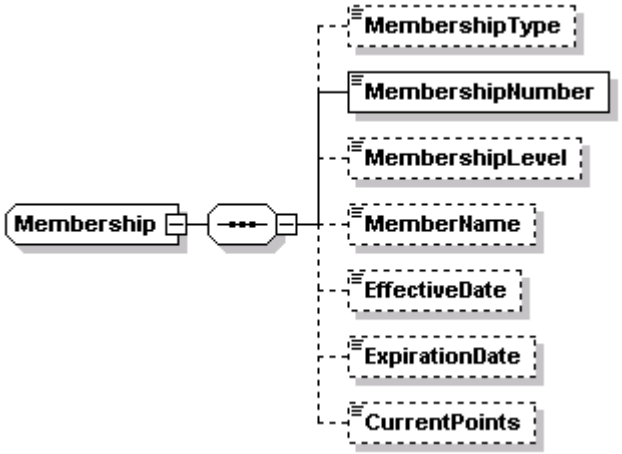


GovernmentIDList

```
<xs:complexType name="GovernmentIDList">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="GovernmentID"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:GovernmentID" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
GovernmentID	element	GovernmentID	optional / multiple	A collection of government ID records.

35.3.2.7 Membership

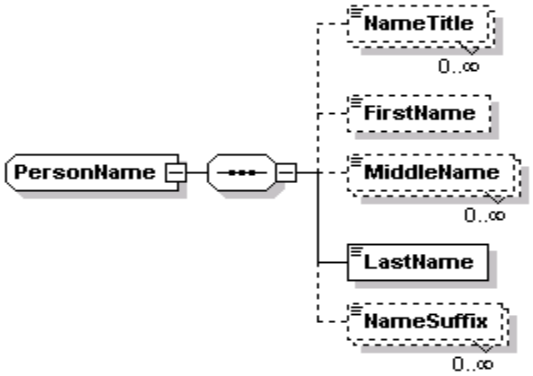


Membership

```
<xs:complexType name="Membership">
    <xs:sequence>
        <xs:element minOccurs="0" name="MembershipType" type="xs:string" />
        <xs:element name="MembershipNumber" type="xs:string" />
        <xs:element minOccurs="0" name="MembershipLevel" type="xs:string" />
        <xs:element minOccurs="0" name="MemberName" type="xs:string" />
        <xs:element minOccurs="0" name="EffectiveDate" type="xs:date" />
        <xs:element minOccurs="0" name="ExpirationDate" type="xs:date" />
        <xs:element minOccurs="0" name="CurrentPoints" type="xs:long" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
MembershipType	element	string	optional	Membership type.
MembershipNumber	element	string	required	Membership number.
MembershipLevel	element	string	optional	Membership level (e.g. GOLD, PLATINUM).
MemberName	element	string	optional	Member name.
EffectiveDate	element	date	optional	Membership effective date (or start date).
ExpirationDate	element	date	optional	Membership expiration date.
CurrentPoints	element	long	optional	Membership point accumulated.

35.3.2.8 PersonName

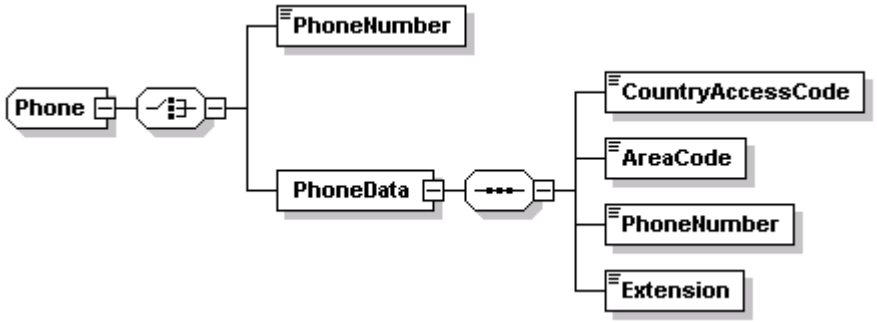


PersonName

```
<xs:complexType name="PersonName">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="NameTitle" type="xs:string" />
        <xs:element minOccurs="0" name="FirstName" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="MiddleName" type="xs:string" />
        <xs:element name="LastName" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="NameSuffix" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="nameOrdered" type="xs:string" />
    <xs:attribute name="familiarName" type="xs:string" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
nameOrdered	attribute	string	none	Not used.
familiarName	attribute	string	none	Familiar name.
NameTitle	element	string	optional / multiple	Name title (e.g. Mr., Mrs., Dr.)
FirstName	element	string	optional	First name.
MiddleName	element	string	optional / multiple	Middle name.
LastName	element	string	required	Last name.
NameSuffix	element	string	optional / multiple	Name suffix (e.g. Jr., III, Esq.)

35.3.2.9 Phone



Phone

```
<xs:complexType name="Phone">
    <xs:choice>
        <xs:element name="PhoneNumber" type="xs:string" />
        <xs:element name="PhoneData">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="CountryAccessCode" type="xs:string" />
                    <xs:element name="AreaCode" type="xs:string" />
                    <xs:element name="PhoneNumber" type="xs:string" />
                    <xs:element name="Extension" type="xs:string" />
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:choice>
    <xs:attribute name="phoneType" type="xs:string" />
    <xs:attribute name="phoneRole" type="xs:string" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
phoneType	attribute	string	none	Type of phone record. Typical values are HOME or BUSINESS.
phoneRole	attribute	string	none	Phone role. Typical values are PHONE, EMAIL, FAX, WEBPAGE.
PhoneNumber	element	string	required	Phone number, email address, web address or other value appropriate to the phone role attribute.
PhoneData	element	PhoneData	required	The phone number may be specified as either a string above, or as individual components as described in the element below

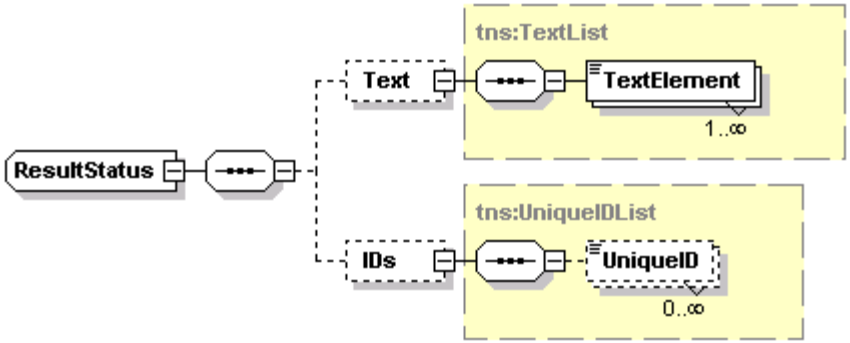
35.3.2.10 PhoneData

PhoneData

```
<xs:element name="PhoneData">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="CountryAccessCode" type="xs:string" />
            <xs:element name="AreaCode" type="xs:string" />
            <xs:element name="PhoneNumber" type="xs:string" />
            <xs:element name="Extension" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Name	Type	Data Type	Use	Comments
CountryAccessCode	element	string	required	Country access code.
AreaCode	element	string	required	Area code.
PhoneNumber	element	string	required	Phone number.
PhoneData	element	Extension	required	Phone extension.

35.3.2.11 ResultStatus



ResultStatus

```
<xs:complexType name="ResultStatus">
    <xs:sequence>
        <xs:element minOccurs="0" name="Text" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:TextList" />
        <xs:element minOccurs="0" name="IDs" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q2:UniqueIDList"
            />
    </xs:sequence>
    <xs:attribute name="resultStatusFlag" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q3:ResultStatusFlag"
        />
```

```
    <xs:attribute name="code" type="xs:string" use="optional" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
resultStatusFlag	attribute	ResultStatusFlag	none	
code	attribute	string	optional	
Text	element	TextList	optional	
IDs	element	UniqueIDList	optional	

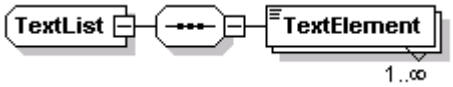
35.3.2.12 Text

Text

```
<xs:complexType name="Text">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="language" type="xs:language" use="optional" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Text	extension	string	optional	Text string.
Language	attribute	language	optional	Language code for Text value.

35.3.2.13 TextList



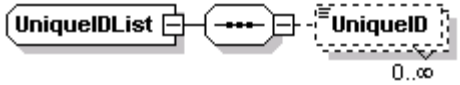
TextList

```
<xs:complexType name="TextList">
    <xs:sequence>
        <xs:element maxOccurs="unbounded" name="TextElement" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q1:Text" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
TextElement	element	Text	required /	

			multiple	

35.3.2.14 UniqueIDList



UniqueIDList

```
<xs:complexType name="UniqueIDList">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="UniqueID"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueID" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
UniqueID	element	UniqueID	optional / multiple	A list of ID values (see below). Records are often identified by a pair of ID's where the source attribute identifies who each ID belongs to.

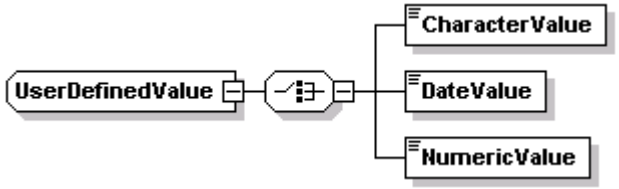
35.3.2.15 UniqueID

UniqueID

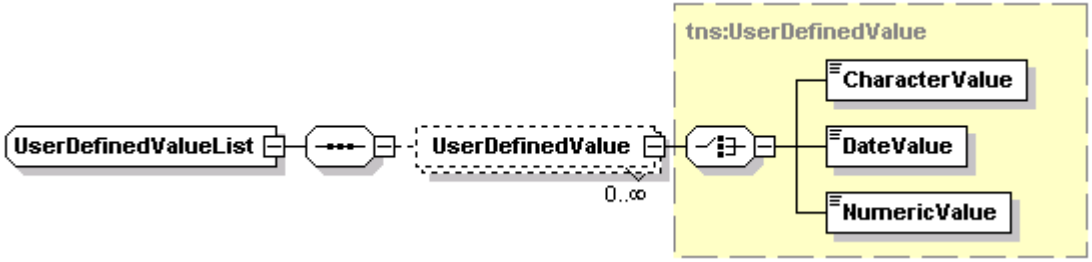
```
<xs:complexType name="UniqueID">
    <xs:simpleContent>
            <xs:extension base="xs:string">
            <xs:attribute name="source" type="xs:string" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
UniqueID	extension	string		The ID value
source	attribute	string	none	The source for the ID. This value identifies the creator/owner of the ID.

35.3.2.16 UserDefinedValue



35.3.2.17 UserDefinedValueList



35.3.2.18 Simple Types

RecordAdministratorAttributes	Attribute Group	insertUser	Internal user code representing the record creator.
		insertDate	Timestamp record was originally created.
		updateUser	Internal user code representing the user who last updated the record
		updateDate	Timestamp when the record was last updated.
		inactiveDate	Timestamp when the record was inactivated. When null, the record is considered active. When set, the record can be treated as though deleted.
BlackListFlag	Simple Type	REMOVE SET	Enumeration of blacklist values.
Gender	Simple Type	FEMALE MALE UNKNOWN	Enumeration for gender.
ReservationStatusType	Simple Type	CANCELLED CHECKED_IN CHECKED_OUT RESERVED WAITLISTED OTHER REVERSE_CHECKED_IN REVERSE_CHECKED_OUT	Enumeration for reservation statuses.
ResultStatusFlag	Simple Type	FAIL SUCCESS	Enumeration for status results.
SubscriptionAction	Simple Type	OTHER SUBSCRIBE	Enumeration of subscription actions to perform.

		UNSUBSCRIBE	

35.3.3 Name Schema Elements (name.xsd)

Namespace	http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types

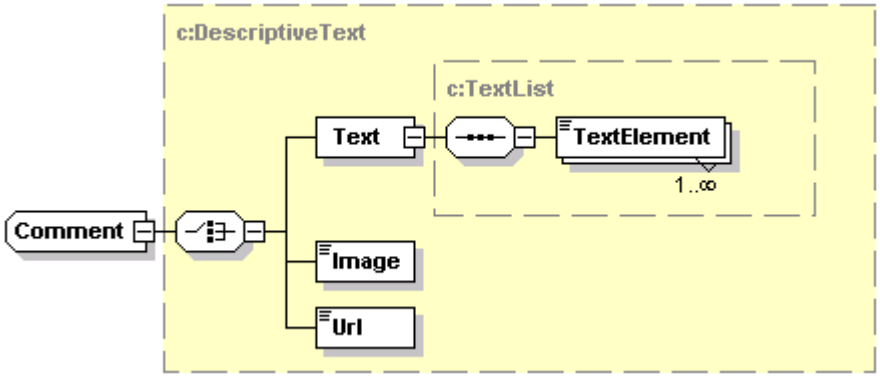
35.3.3.1 BlackList

BlackList

```
<xs:complexType name="BlackList">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="flag" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:BlackListFlag" />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
BlackList	extension	string		Blacklist flag.
flag	attribute	BlackListFlag	none	One of *REMOVE* or *SET* indicating whether the blacklist flag should be set or not.

35.3.3.2 Comment



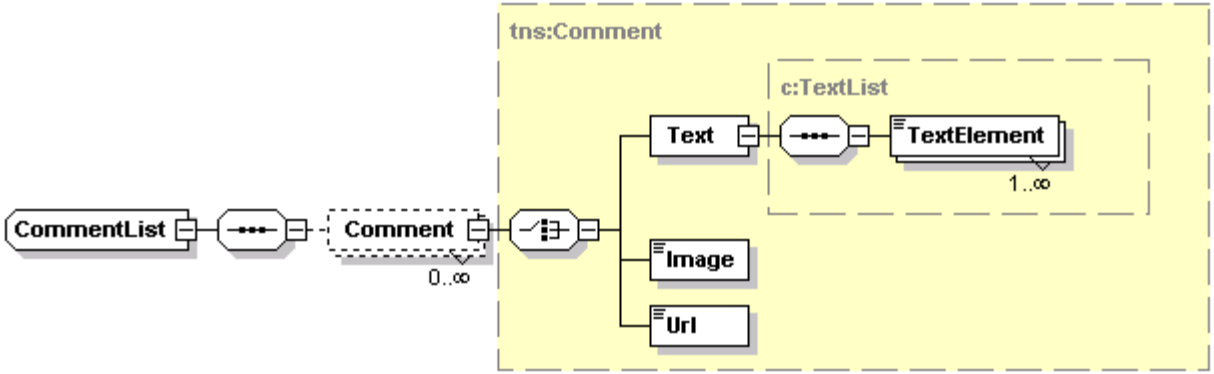
Comment

```
<xs:complexType name="Comment">
    <xs:complexContent mixed="false">
        <xs:extension xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" base="q1:DescriptiveText">
            <xs:attribute name="commentType" type="xs:string" />
```

```
        <xs:attributeGroup ref="q1:RecordAdministratorAttributes" />
      </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Comment	extension	DescriptiveText		Extension of DescriptiveText
commentType	attribute	string	none	Type of comment.
RecordAdministratorAttributes	attr group			Record creation and last update attributes.

35.3.3.3 CommentList

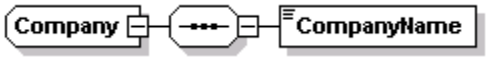


CommentList

```
<xs:complexType name="CommentList">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Comment"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:Comment" />
    </xs:sequence>
</xs:complexType>
```

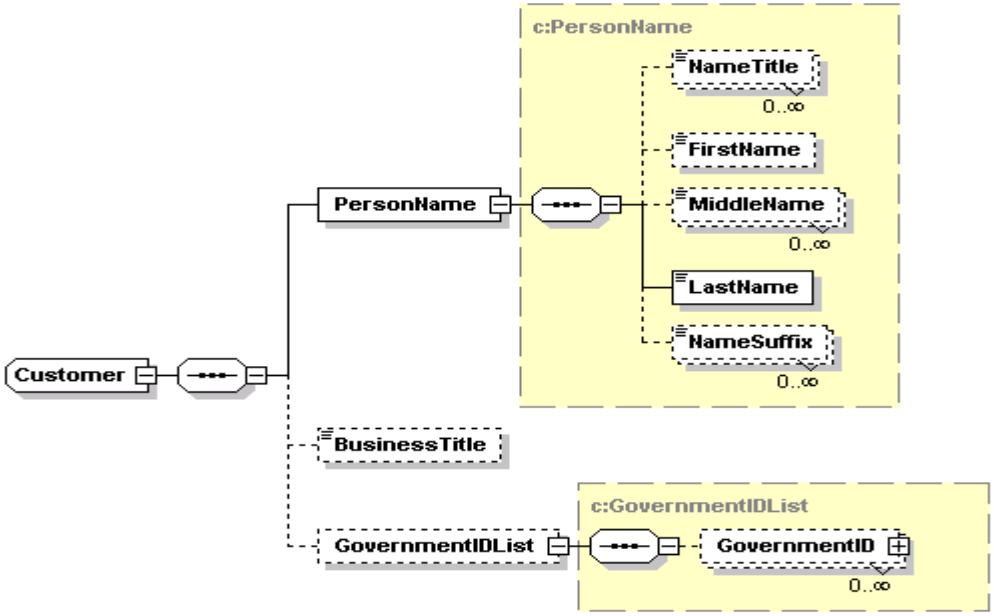
Name	Type	Data Type	Use	Comments
Comment	element	Comment	optional / multiple	A collection of Comment elements.

35.3.3.4 Company



Company				
<xs:complexType name="Company">
<xs:sequence>
<xs:element name="CompanyName" type="xs:string" />
</xs:sequence>
<xs:attribute name="commissionCode" type="xs:string" />
</xs:complexType>				
Name	**Type**	**Data Type**	**Use**	**Comments**
commissionCode	attribute	string	none	Not used.
CompanyName	element	string	required	Company name.

35.3.3.5 Customer

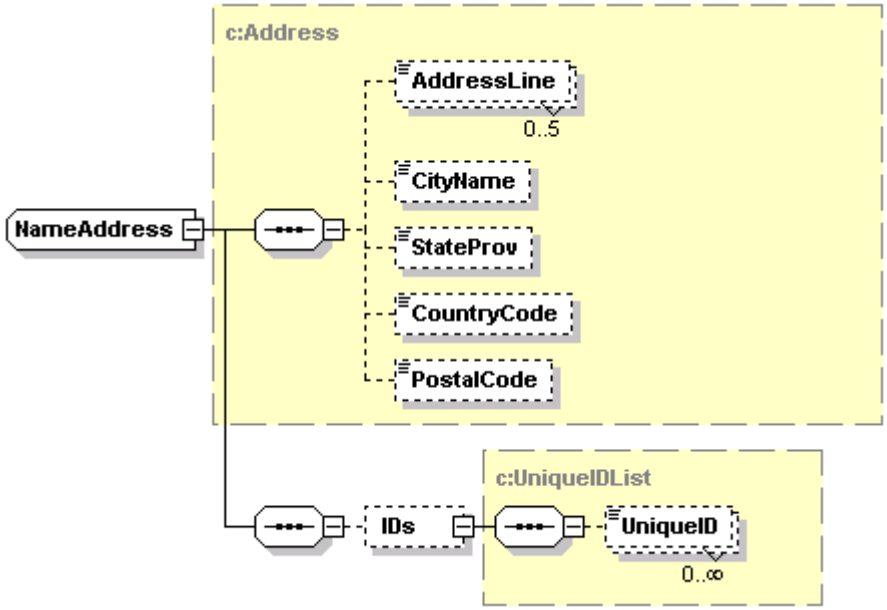


Customer
<xs:complexType name="Customer">>

```
    <xs:sequence>
        <xs:element name="PersonName" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:PersonName" />
        <xs:element minOccurs="0" name="BusinessTitle" type="xs:string" />
        <xs:element minOccurs="0" name="GovernmentIDList" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q2:GovernmentIDList" />
    </xs:sequence>
    <xs:attribute name="gender" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q3:Gender" />
    <xs:attribute name="birthDate" type="xs:date" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
gender	attribute	Gender	none	The guest's gender.
birthDate	attribute	date	none	The guest's birth date.
PersonName	element	PersonName	required	The guest name record.
BusinessTitle	element	string	optional	The guest's title.
GovernmentIDList	element	GovernmentIDList	optional	A collection of government ID's.

35.3.3.6 NameAddress

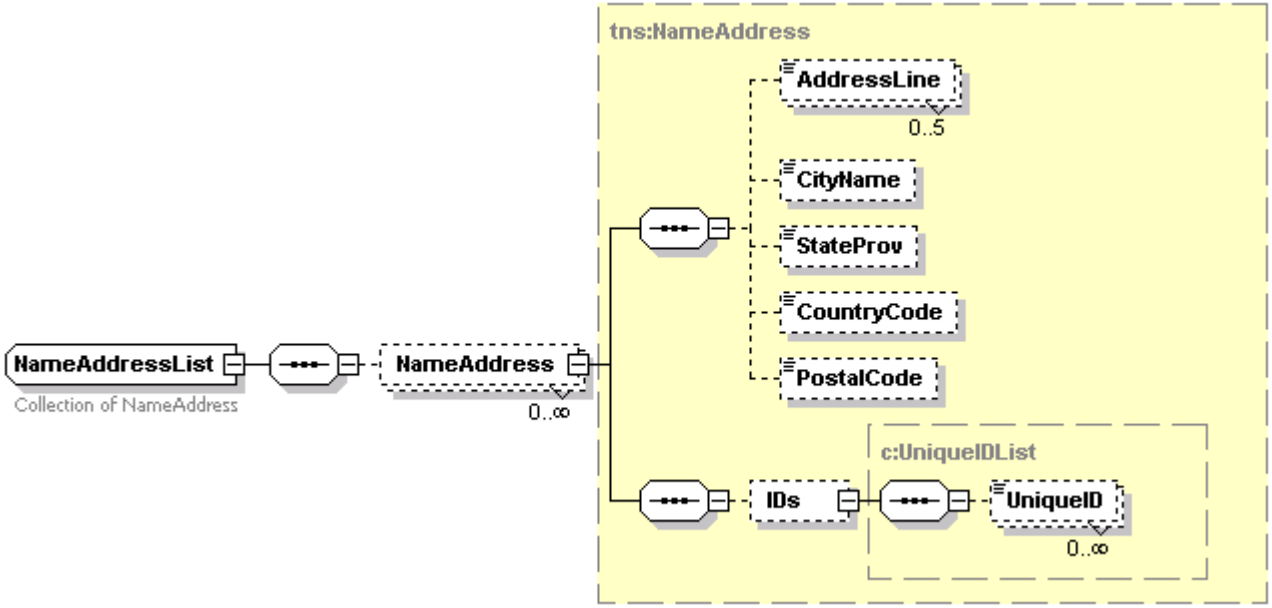


NameAddress

```
<xs:complexType name="NameAddress">
    <xs:complexContent mixed="false">
        <xs:extension xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" base="q1:Address">
            <xs:sequence>
                <xs:element minOccurs="0" name="IDs" type="q1:UniqueIDList" />
            </xs:sequence>
            <xs:attribute name="primary" type="xs:boolean" />
            <xs:attribute name="displaySequence" type="xs:int" />
            <xs:attributeGroup ref="q1:RecordAdministratorAttributes" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
NameAddress	extension	Address		Extension of Address element type.
primary	attribute	boolean	none	Flag to indicate primary address.
displaySequence	attribute	int	none	Display sequence.
RecordAdministratorAttributes	attr group			Record creation and last update attributes.
IDs	element	UniqueIDList	optional	Internal ID values.

35.3.3.7 NameAddressList

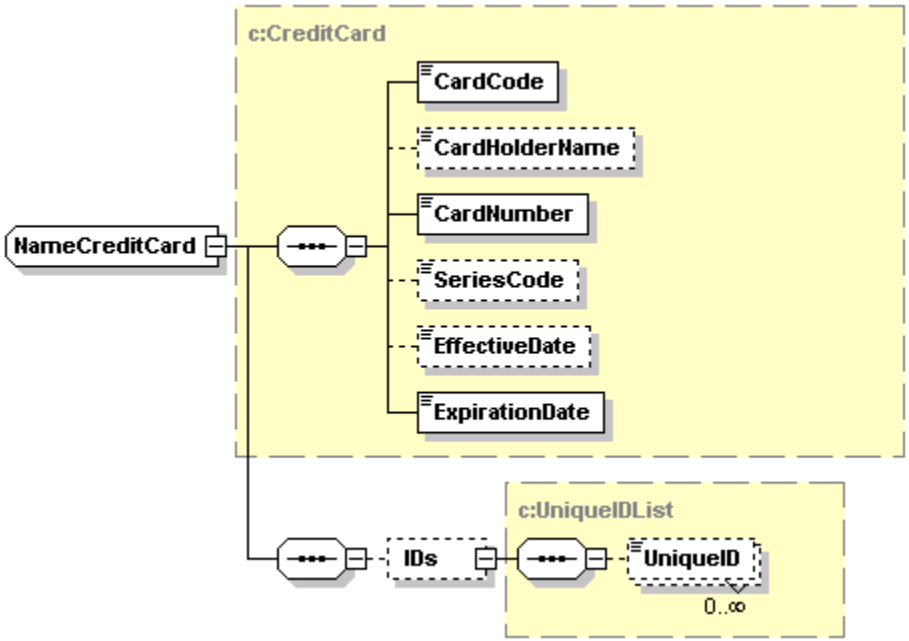


NameAddressList

```
<xs:complexType name="NameAddressList">
    <xs:annotation>
    <xs:documentation>Collection of NameAddress</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="NameAddress"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:NameAddress" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
NameAddress	element	NameAddress	optional / multiple	A collection of NameAddress records.

35.3.3.8 NameCreditCard



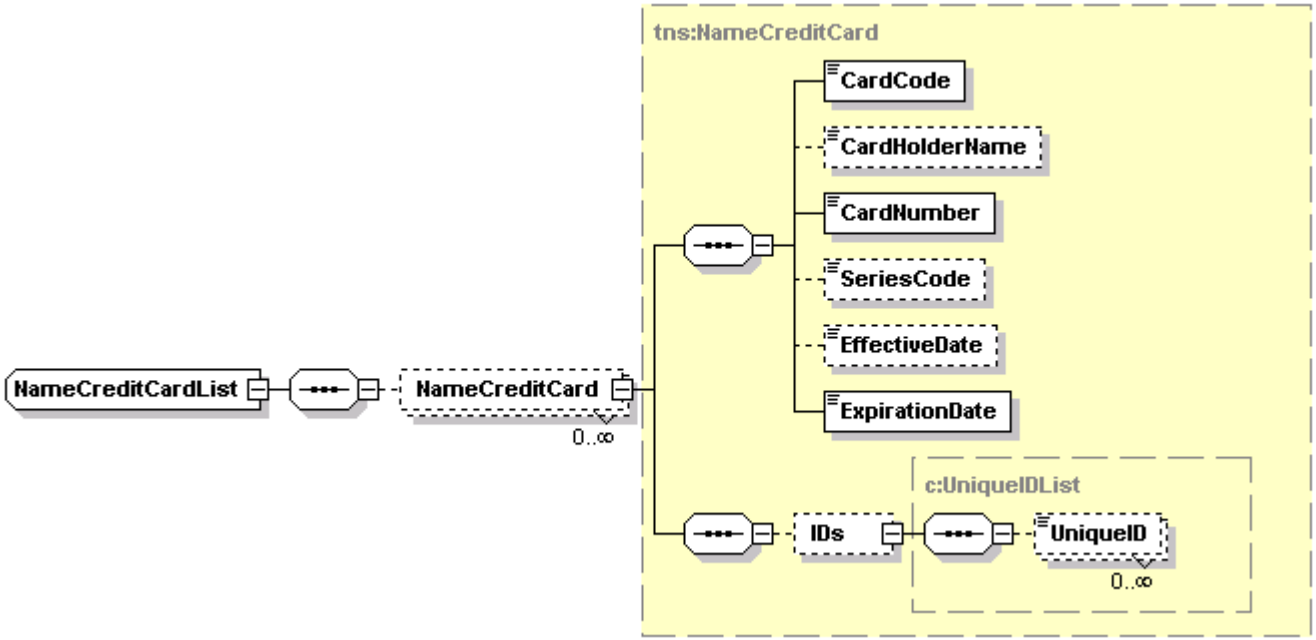
NameCreditCard

```
<xs:complexType name="NameCreditCard">
    <xs:complexContent mixed="false">
        <xs:extension xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" base="q1:CreditCard">
            <xs:sequence>
                <xs:element minOccurs="0" name="IDs" type="q1:UniqueIDList" />
            </xs:sequence>
            <xs:attribute name="primary" type="xs:boolean" />
            <xs:attribute name="displaySequence" type="xs:int" />
            <xs:attributeGroup ref="q1:RecordAdministratorAttributes" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
NameCreditCard	extension	CreditCard		Extension of CreditCard element.
primary	attribute	boolean	none	Flag to indicate primary credit card.

displaySequence	attribute	int	none	Display order.
RecordAdministratorAttributes	attr group			Record creation and last update attributes.
IDs	element	UniqueIDList	optional	Internal ID's.

35.3.3.9 NameCreditCardList

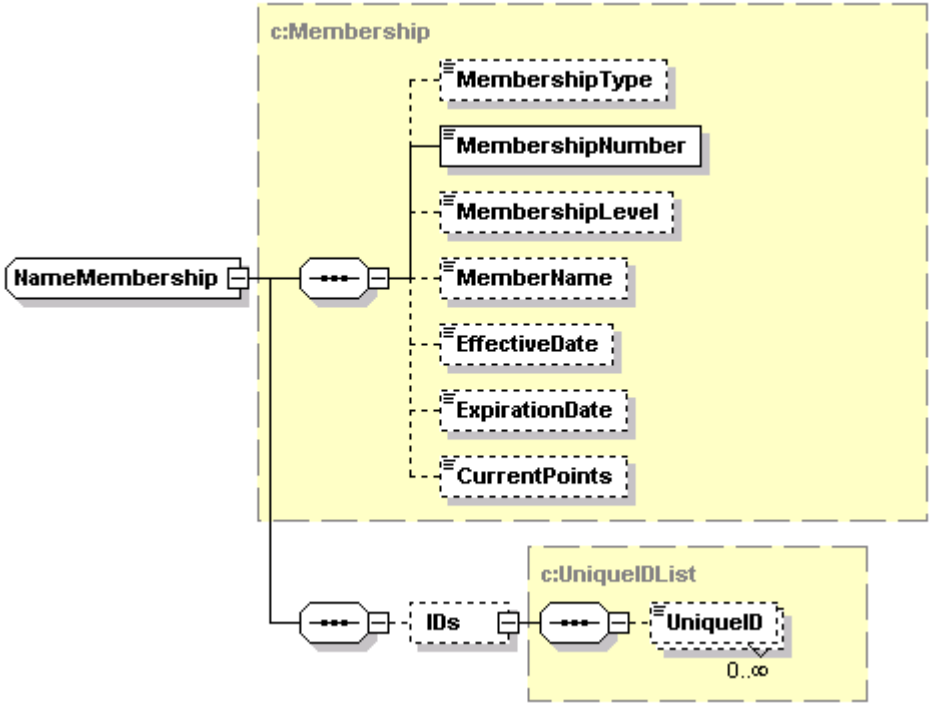


NameCreditCardList

```
<xs:complexType name="NameCreditCardList">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="NameCreditCard"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:NameCreditCard" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
NameCreditCard	element	NameCreditCard	optional / multiple	A collection of credit card elements.

35.3.3.10 NameMembership

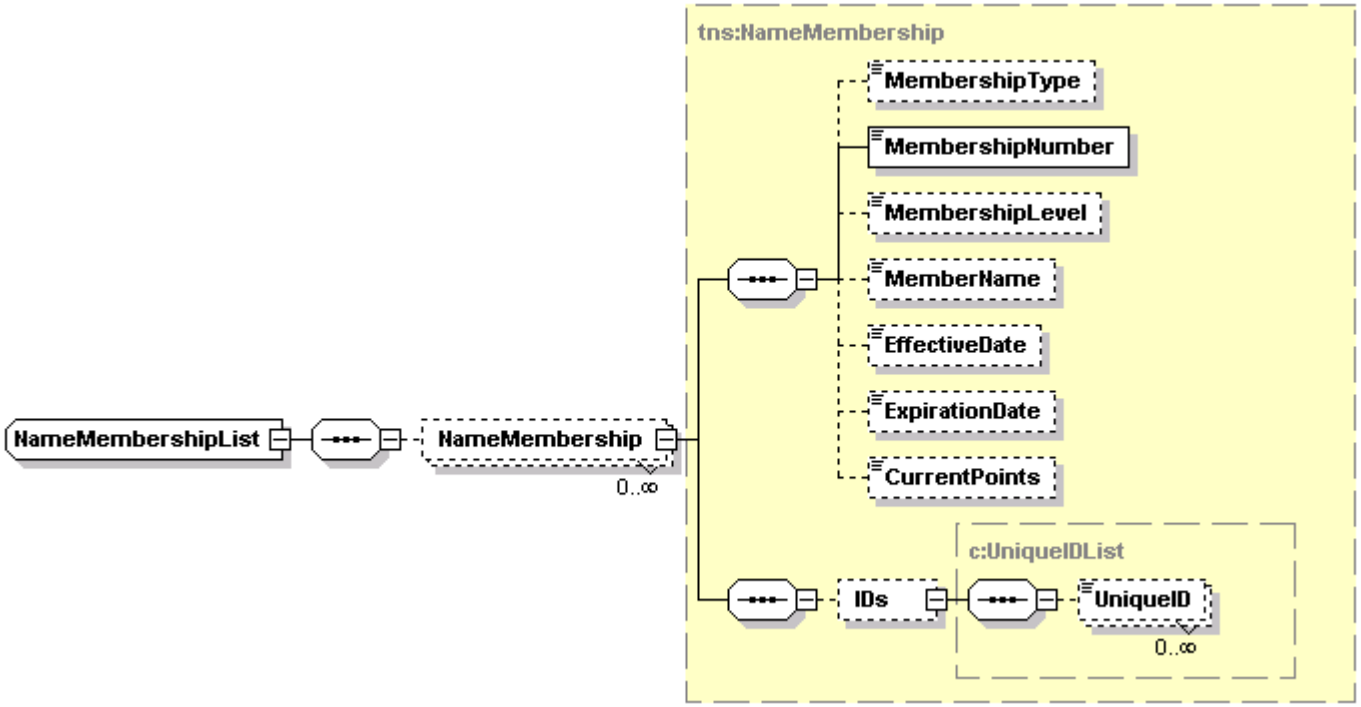


NameMembership

```
<xs:complexType name="NameMembership">
    <xs:complexContent mixed="false">
        <xs:extension xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" base="q1:Membership">
            <xs:sequence>
                <xs:element minOccurs="0" name="IDs" type="q1:UniqueIDList" />
            </xs:sequence>
            <xs:attribute name="primary" type="xs:boolean" />
            <xs:attribute name="displaySequence" type="xs:int" />
            <xs:attributeGroup ref="q1:RecordAdministratorAttributes" />
        </xs:extension>
    </xs:complexContent>
```

</xs:complexType>				
Name	**Type**	**Data Type**	**Use**	**Comments**
NameMembership	extension	Membership		Extension of Membership element.
primary	attribute	boolean	none	Flag to indicate primary membership.
displaySequence	attribute	int	none	Display sequence.
RecordAdministratorAttributes	attr group			Record creation and last update attributes.
IDs	element	UniqueIDList	optional	Internal record ID's.

35.3.3.11 NameMembershipList



NameMembershipList
<xs:complexType name="NameMembershipList"> <xs:sequence>

```
        <xs:element minOccurs="0" maxOccurs="unbounded" name="NameMembership"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:NameMembership" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
NameMembership	element	NameMembership	optional / multiple	A collection of NameMembership elements.

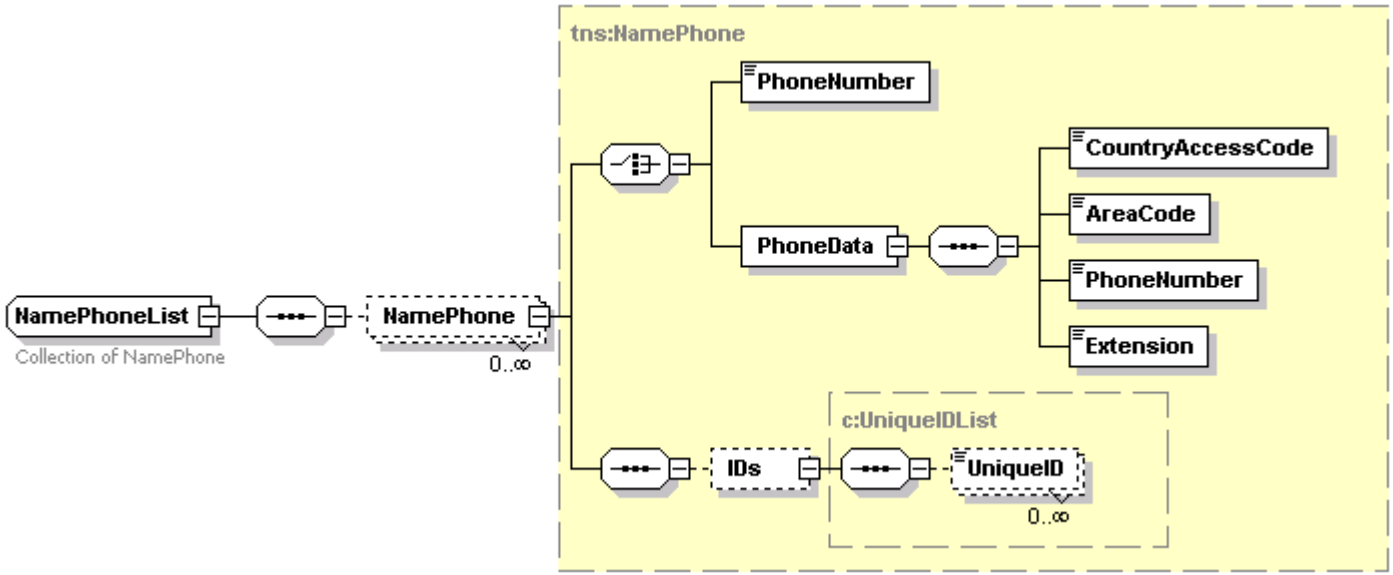
35.3.3.12 NamePhone

NamePhone

```
<xs:complexType name="NamePhone">
    <xs:complexContent mixed="false">
        <xs:extension xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" base="q1:Phone">
            <xs:sequence>
                <xs:element minOccurs="0" name="IDs" type="q1:UniqueIDList" />
            </xs:sequence>
            <xs:attribute name="primary" type="xs:boolean" />
            <xs:attribute name="displaySequence" type="xs:int" />
            <xs:attributeGroup ref="q1:RecordAdministratorAttributes" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
NamePhone	extension	Phone		Extension of Phone element type.
primary	attribute	boolean	none	Flag to indicate primary phone.
displaySequence	attribute	int	none	Display sequence.
RecordAdministratorAttributes	attr group			Record creation and update attributes.
IDs	element	UniqueIDList	optional	Internal record ID's.

35.3.3.13 NamePhoneList

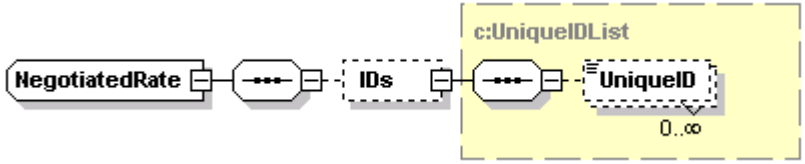


NamePhoneList

```
<xs:complexType name="NamePhoneList">
    <xs:annotation>
    <xs:documentation>Collection of NamePhone</xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="NamePhone"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:NamePhone" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
NamePhone	element	NamePhone	optional / multiple	A collection of NamePhone records.

35.3.3.14 NegotiatedRate

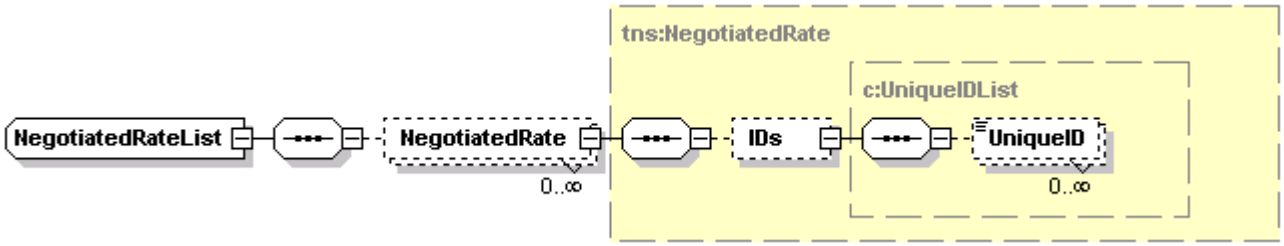


NegotiatedRate

```
<xs:complexType name="NegotiatedRate">
    <xs:sequence>
        <xs:element minOccurs="0" name="IDs" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueIDList"
            />
    </xs:sequence>
    <xs:attribute name="resortCode" type="xs:string" />
    <xs:attribute name="rateCode" type="xs:string" use="required" />
    <xs:attribute name="commissionCode" type="xs:string" />
    <xs:attribute name="beginDate" type="xs:date" />
    <xs:attribute name="endDate" type="xs:date" />
    <xs:attribute name="displaySequence" type="xs:int" />
    <xs:attributeGroup xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" ref="q2:RecordAdministratorAttributes" />
</xs:complexType>
```

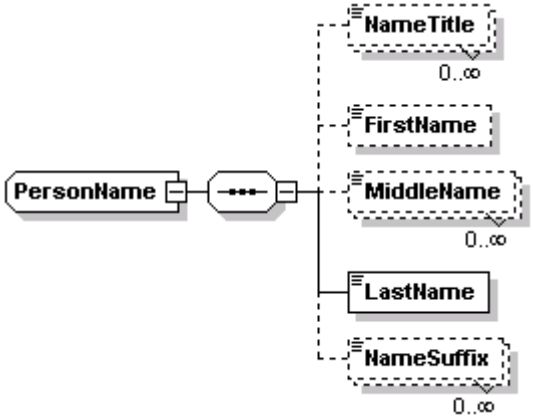
Name	Type	Data Type	Use	Comments
resortCode	attribute	string	none	Property code where negotiated rate is applicable.
rateCode	attribute	string	required	Rate code.
commissionCode	attribute	string	none	Commission code.
beginDate	attribute	date	none	Begin date for rate.
endDate	attribute	date	none	End date for rate.
displaySequence	attribute	int	none	Display sequence.
RecordAdministratorAttributes	attr group			Record creation and last update attributes.
IDs	element	UniqueIDList	optional	Internal record ID's.

35.3.3.15 NegotiatedRateList



NegotiatedRateList				
<xs:complexType name="NegotiatedRateList"> <xs:sequence> <xs:element minOccurs="0" maxOccurs="unbounded" name="NegotiatedRate" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:NegotiatedRate" /> </xs:sequence> </xs:complexType>				
Name	**Type**	**Data Type**	**Use**	**Comments**
NegotiatedRate	element	NegotiatedRate	optional / multiple	A collection of NegotiatedRate elements.

35.3.3.16 PersonName

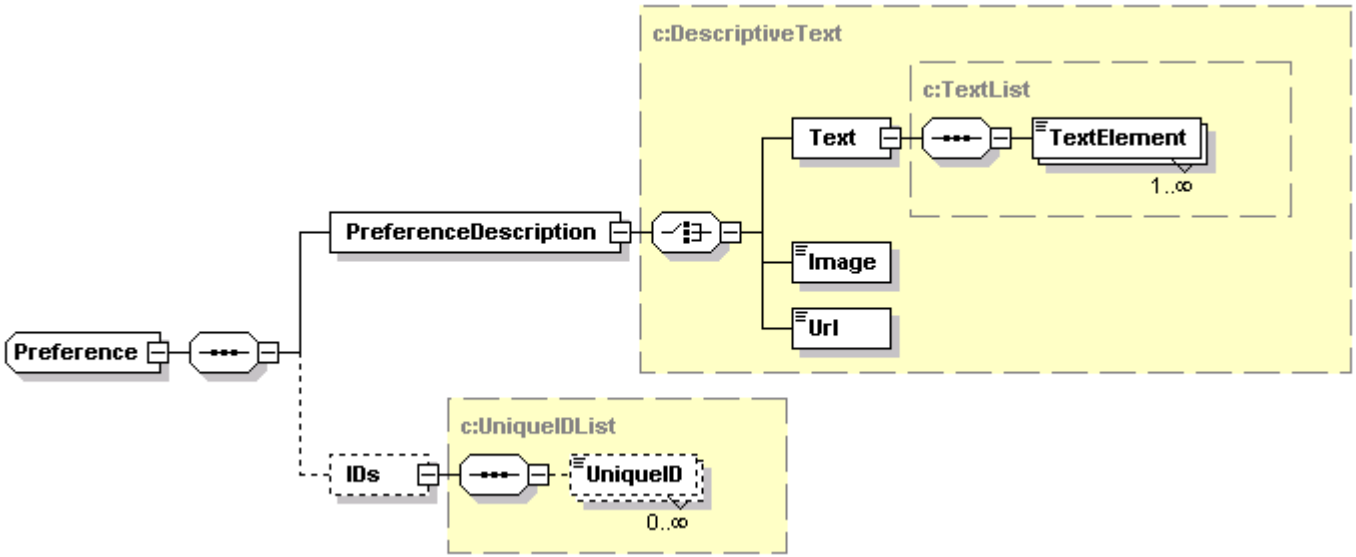


PersonName

```
<xs:complexType name="PersonName">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="NameTitle" type="xs:string" />
        <xs:element minOccurs="0" name="FirstName" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="MiddleName" type="xs:string" />
        <xs:element name="LastName" type="xs:string" />
        <xs:element minOccurs="0" maxOccurs="unbounded" name="NameSuffix" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="nameOrdered" type="xs:string" />
    <xs:attribute name="familiarName" type="xs:string" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
nameOrdered	attribute	string	none	The ordered representation of the guest name.
familiarName	attribute	string	none	The guest's familiar name.
NameTitle	element	string	optional / multiple	Name title such as "Mr.", "Mrs.", or "Dr."
FirstName	element	string	optional	The guest's first name.
MiddleName	element	string	optional / multiple	The guest's middle name(s).
LastName	element	string	required	The guest's last name. This is the only field which is required.
NameSuffix	element	string	optional / multiple	Name suffix values such as "Jr.", "III", "Esq."

35.3.3.17 Preference



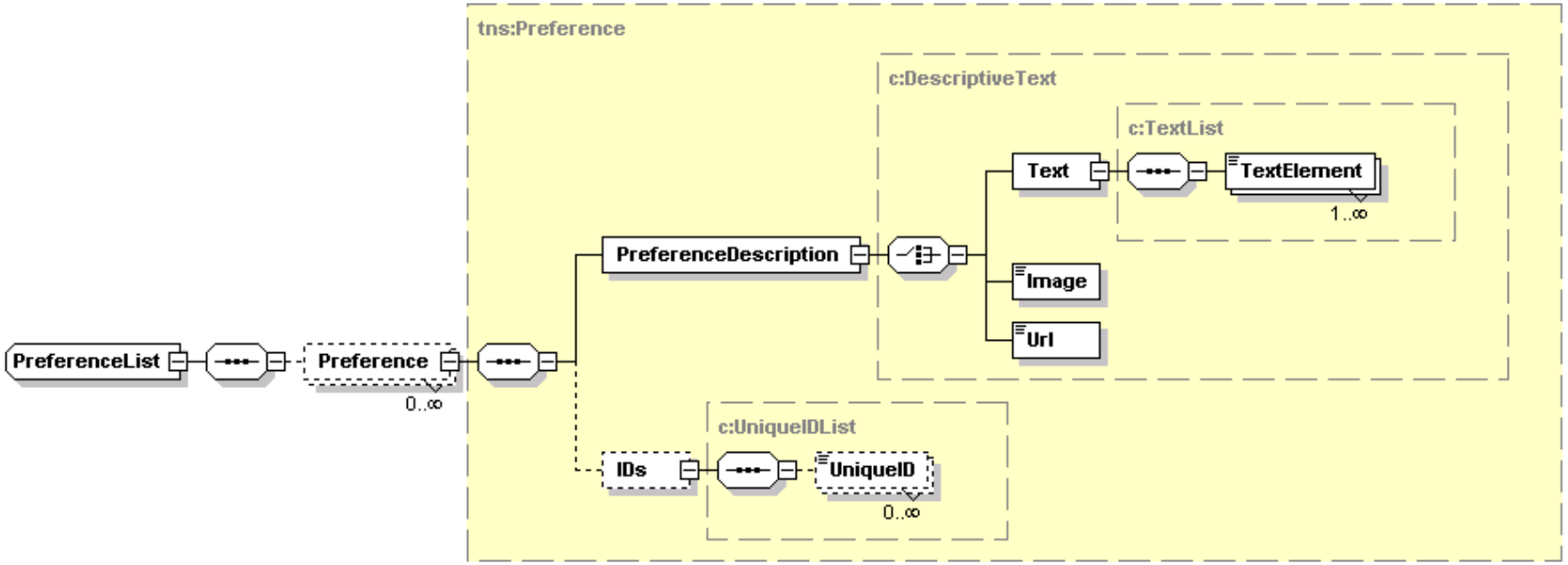
Preference

```
<xs:complexType name="Preference">
    <xs:sequence>
        <xs:element name="PreferenceDescription" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q1:DescriptiveText" />
        <xs:element minOccurs="0" name="IDs" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q2:UniqueIDList"
            />
    </xs:sequence>
    <xs:attribute name="resortCode" type="xs:string" />
    <xs:attribute name="preferenceType" type="xs:string" />
    <xs:attribute name="otherPreferenceType" type="xs:string" />
    <xs:attribute name="preferenceValue" type="xs:string" />
    <xs:attributeGroup xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" ref="q3:RecordAdministratorAttributes" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
resortCode	attribute	string	none	Property code where preference is defined.
preferenceType	attribute	string	none	Preference type code.
otherPreferenceType	attribute	string	none	Not used.
preferenceValue	attribute	string	none	Preference value.

RecordAdministratorAttributes	attr group			Record creation and last update attributes.
PreferenceDescription	element	DescriptiveText	required	Description of the preference.
IDs	element	UniqueIDList	optional	Internal record ID's.

35.3.3.18 PreferenceList

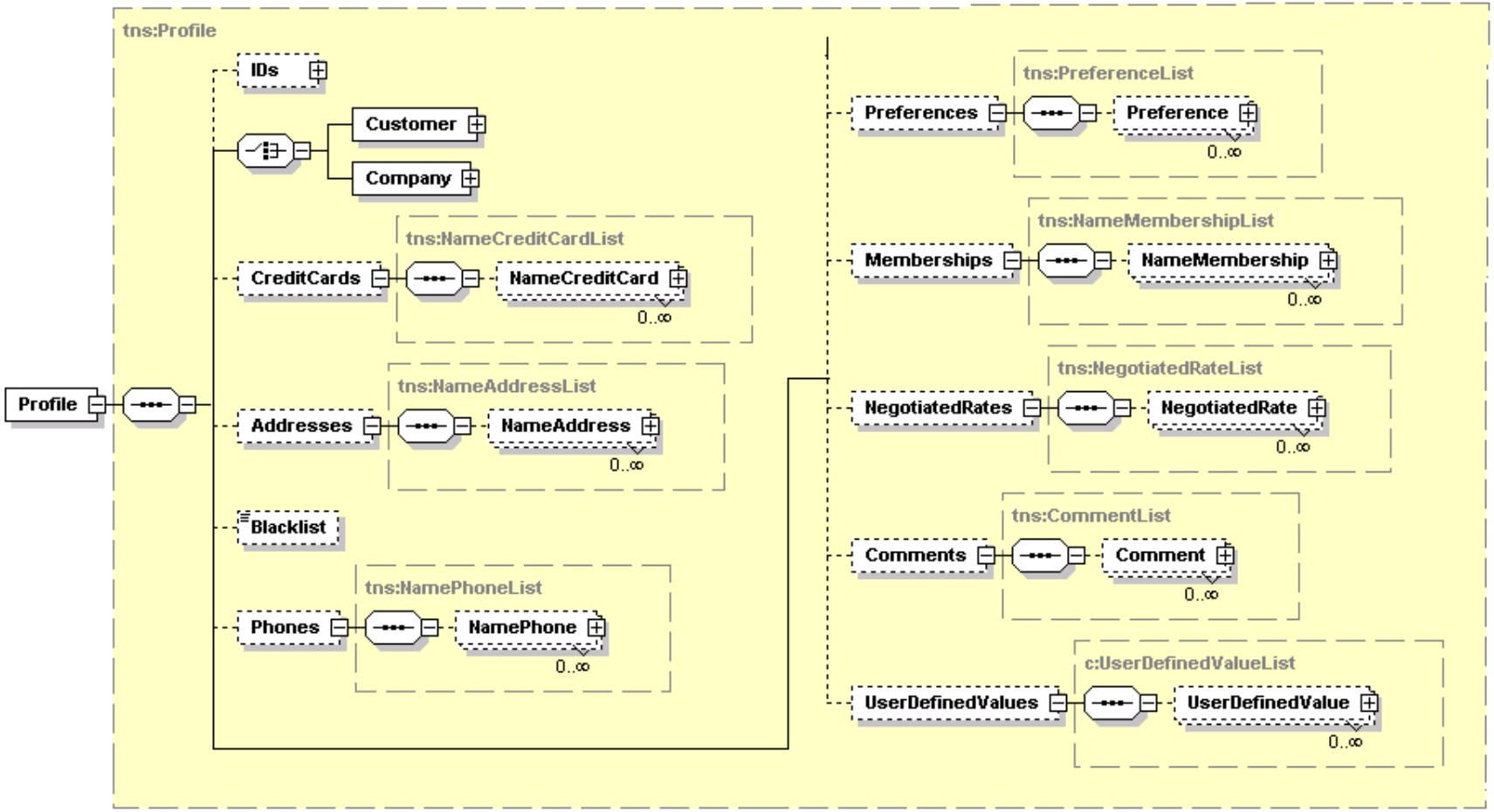


PreferenceList

```
<xs:complexType name="PreferenceList">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Preference"
            xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q1:Preference" />
    </xs:sequence>
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
Preference	element	Preference	optional / multiple	A collection of Preference elements.

35.3.3.19 Profile



Profile

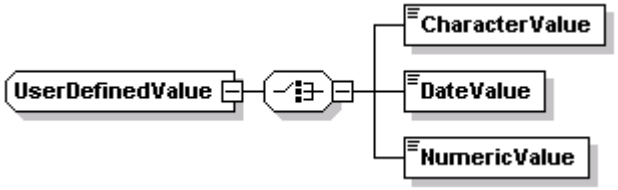
```
<xs:complexType name="Profile">
    <xs:sequence>
        <xs:element minOccurs="0" name="IDs" xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UniqueIDList"
```

```
            />
        <xs:choice>
            <xs:element name="Customer" xmlns:q2="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q2:Customer" />
            <xs:element name="Company" xmlns:q3="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q3:Company" />
        </xs:choice>
        <xs:element minOccurs="0" name="CreditCards" xmlns:q4="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
            type="q4:NameCreditCardList" />
        <xs:element minOccurs="0" name="Addresses" xmlns:q5="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
            type="q5:NameAddressList" />
        <xs:element minOccurs="0" name="Blacklist" xmlns:q6="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types" type="q6:BlackList" />
        <xs:element minOccurs="0" name="Phones" xmlns:q7="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
            type="q7:NamePhoneList" />
        <xs:element minOccurs="0" name="Preferences" xmlns:q8="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
            type="q8:PreferenceList" />
        <xs:element minOccurs="0" name="Memberships" xmlns:q9="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
            type="q9:NameMembershipList" />
        <xs:element minOccurs="0" name="NegotiatedRates" xmlns:q10="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
            type="q10:NegotiatedRateList" />
        <xs:element minOccurs="0" name="Comments" xmlns:q11="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Name/Types"
            type="q11:CommentList" />
        <xs:element minOccurs="0" name="UserDefinedValues" xmlns:q12="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types"
            type="q12:UserDefinedValueList" />
    </xs:sequence>
    <xs:attribute name="nameType" type="xs:string" />
    <xs:attribute name="protected" type="xs:boolean" />
    <xs:attribute name="languageCode" type="xs:string" />
    <xs:attribute name="nationality" type="xs:string" />
    <xs:attribute name="vipCode" type="xs:string" />
    <xs:attribute name="taxExempt" type="xs:boolean" />
    <xs:attributeGroup xmlns:q13="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" ref="q13:RecordAdministratorAttributes" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
nameType	attribute	string	none	Name type. Typical values are *GUEST* or *COMPANY*.
protected	attribute	boolean	none	Flag to indicated whether profile can be modified by external system.
languageCode	attribute	string	none	Default language code of guest.
nationality	attribute	string	none	Nationality of guest.
vipCode	attribute	string	none	VIP value.
taxExempt	attribute	boolean	none	Not used.
RecordAdministratorAttributes	attr group			Record of creation and last change.
IDs	element	UniqueIDList	optional	Internal profile ID.
Customer	element	Customer	required	The guest name. Either this element or the Company element must be defined. The minimum requirement is the guest's last name.
Company	element	Company	required	The company name. Either this element of the Customer element must

				be defined.
CreditCards	element	NameCreditCardList	optional	A collection of credit card elements.
Addresses	element	NameAddressList	optional	A collection of address records.
Blacklist	element	BlackList	optional	A flag to indicate whether the guest has been blacklisted.
Phones	element	NamePhoneList	optional	A collection of telephone elements.
Preferences	element	PreferenceList	optional	A collection of guest preference elements.
Memberships	element	NameMembershipList	optional	A collection of membership elements.
NegotiatedRates	element	NegotiatedRateList	optional	A collection of negotiated rate elements.
Comments	element	CommentList	optional	Comments.
UserDefinedValues	element	UserDefinedValueList	optional	A collection of customizable system defined values.

35.3.3.20 UserDefinedValue

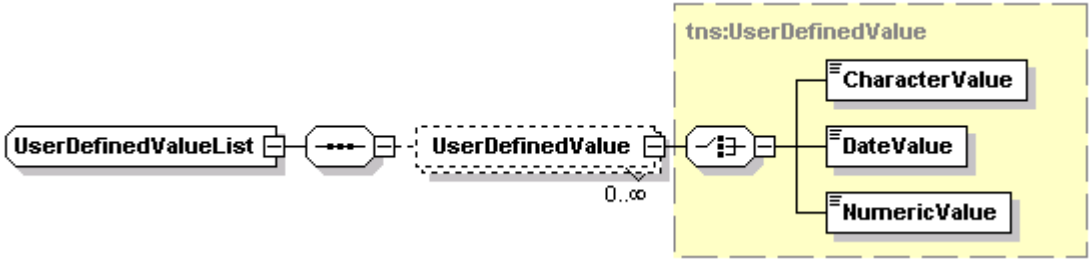


UserDefinedValue

```
<xs:complexType name="UserDefinedValue">
    <xs:choice>
        <xs:element name="CharacterValue" type="xs:string" />
        <xs:element name="DateValue" type="xs:dateTime" />
        <xs:element name="NumericValue" type="xs:float" />
    </xs:choice>
    <xs:attribute name="valueName" type="xs:string" />
</xs:complexType>
```

Name	Type	Data Type	Use	Comments
valueName	attribute	string	none	User defined value name.
CharacterValue	element	string	required	Character based value (Choice).
DateValue	element	dateTime	required	Date value (Choice).
NumericValue	element	float	required	Numeric value (Choice).

35.3.3.21 UserDefinedValueList



UserDefinedValueList				
<xs:complexType name="UserDefinedValueList">
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="unbounded" name="UserDefinedValue"
xmlns:q1="http://htng.org/PWSWG/2006/08/SingleGuestItinerary/Common/Types" type="q1:UserDefinedValue" />
</xs:sequence>
</xs:complexType>				
Name	**Type**	**Data Type**	**Use**	**Comments**
UserDefinedValue	element	UserDefinedValue	optional / multiple	A collection of user defined values.

Note: This specification is included for information and reference only. It is not included in the Certification Program at this time. It is subject to change in the future.

Part 6 Guest Self Service Version 1.0

Chapter 36 Document History

36.1 Document Changes

Version	Date	Author	Comments
0.1			Initial submission
0.2			Added template types: Activity, Dining, Retail, GiftCertificate Unified and extended messages: TemplateCategory, TemplateInfo, Availability, Sale, FetchBooking
0.3			New message: PaymentCondition for prepayments and cancellations
0.4			Profile management by SGI spec, remove header of each example, minor changes (Attribute names, ..)
0.5			Add table to each message for details like Data Type, Use, Comments, ... Add global schema elements. TemplateCategory: Rank -> Order TemplateInfo: One generic template with type attribute, Prices -> optional, Duration -> optional, MediaPath -> MediaURI, MediaFormat: MIME type PaymentCondition: New element Value with attribute type, instead of FixPrice and Percentage element. unit Attribute for: CancellationWindow, Duration, StartInterval,MinimumInAdvance,MaximumInAdvance
0.6			New self service frontend roles
0.7			Use datatype duration instead of TimeUnitType TemplateInfo: Rename Links to RelatedItems and new relatedItemType added; Multiple taxes added PaymentCondition: DueDate added; two new payment condition types: FULLPAYMENT, BALANCEPAYMENT; Taxes added; Currency for payment condition value added Availability: Use of generic preferences with several types; availabilityType added; Complete profile instead of profileIDs.
0.8			Create request: Add PaymentGuarantee -> Credit card or reservation
0.9			Sale and Create: Use PaymentIdentification element for Create(PaymentGuarantee) and for Sale (PaymentInformation). Sale: Description text for ShippingCostItem element. Sale response: ReceiptImage added. BookingIdentification element: Remark added.
0.91			TemplateInfo: New template type SHIPPINGCOSTS. Sale: TemplateID to ShippingCostItem added. Availability/Create: Element ModifyBookingIDs added to modify existing bookings.

			Changes to FetchBookingRequest/Response (Additional filter, Preferences, PreferredResources, ...)
0.92			TemplateInfo: Element Note added for Instructions Availability element: Price -> optional for dining FetchBookingResponse: Optional element Remark added GiftCertificateIdentification: VerificationCode added New message: FetchGiftCertificate
0.93			New URI: http://htng.org/PWS/2008A/GuestSelfService
1.0			Conversion to HTNG Template

Chapter 37 Document Information

37.1 Overview

Global schema elements used in more than one message are included in the Global Schema Elements section of this document and marked with green color.

All data types, which are defined in the Single Guest Itinerary specification and used in this document, are marked with blue color and listed here:

- Profile
- PersonName
- Text
- Amount
- UniqueIDList
- ResultStatus
- OGTimeSpan
- PersonName
- NameMembership

37.2 Document Terms

For the purpose of this document the following terms have been defined as follows:

Term	Definition
SSF	Self Service Frontend
Roles:	
WS	Web site
HSIA	High speed internet access
VOD	Video on demand
KIOSK	Kiosk
CMS	Content managing system
SSS	Services Scheduling System

37.3 Referenced Documents

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

Name	Location
HTNG Single Guest Itinerary Specification	

Chapter 38 HTTP/SOAP

Communication between SSF and SSS takes place through HTTP requests and their responses.

The XML data structure corresponds to the structure used with permanent connection but with the data being enclosed in a SOAP envelope.
All messages must include a soap header that conforms to the WSAddressing and WSSecurity specifications. Receiving systems may optionally require that the wsa:To element match a specific destination address, and therefore, this element should be configurable at run time. In addition, all sending systems must identify themselves by specifying a wsa:From element unique for their system. Typically, this is in the form of "URN:<system>". The wsa:ReplyTo address may be the anonymous form described in the August, 2004 specification.

WSSecurity may be optionally implemented. Providers are only required to support basic plain text authentication with a username and password. The user credentials will be limited to a single login per system, agreed between two vendors. There is no requirement to support multiple user logins from a single vendor through this interface.

Sample Header Message

```
<soap:Header>
        <wsa:Action>http://htng.org/PWS/2008A/GuestSelfService#TemplateCategory</wsa:Action>
        <wsa:MessageID>urn:uuid:11f46e88-8d20-6e65-3812-da0a786a90ef</wsa:MessageID>
        <wsa:ReplyTo>
                <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
        </wsa:ReplyTo>
        <wsa:To>urn:TAC</wsa:To>
        <wsa:From>
                <wsa:Address>urn:ES</wsa:Address>
        </wsa:From>
        <wsse:Security soap:mustUnderstand="1">
        <wsu:Timestamp wsu:Id="Timestamp-97134fd8-7d8f-bc2c-14d0-9e8197fdc7e9">
                <wsu:Created>2007-05-14T07:51:15Z</wsu:Created>
                <wsu:Expires>2007-05-14T08:51:15Z</wsu:Expires>
        </wsu:Timestamp>
        <wsse:UsernameToken wsu:Id="SecurityToken-02115bec-7c32-1f79-c71c-0045929720ff">
                <wsse:Username>HTNG</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
        1.0#PasswordText">HTNG</wsse:Password>
                <wsse:Nonce>YymhHIFQrrxNkmid6cMrJQ==</wsse:Nonce>
                <wsse:Created>2007-05-14T07:51:15Z</wsse:Created>
        </wsse:UsernameToken>
        </wsse:Security>
</soap:Header>
```

All further examples are shown space saving, without SOAP envelope and without namespace declarations.

Chapter 39 Profile Management

(SSS ⇔ SSF)

The Guest Self Service interface uses the already defined profile management of the Single Guest Itinerary specification.
To avoid duplicate guest profiles, the UpdateProfile request can be used without ProfileIDs. When the ProfileIDs element is missing, the receiving system has to check by lastname, firstname, etc., if the guest already exists or not.

For further details we refer to the SGI specification.

Chapter 40 Message Definitions

40.1 Definition TemplateCategory

(SSF ⇨ SSS)

This request is used to retrieve template categories. The template categories can be used to group the templates in the SSF.

Soap Action http://htng.org/PWS/2008A/GuestSelfService#TemplateCategory

Request:

TemplateCategoryRequest				
Name	**Type**	**Data Type**	**Use**	**Comments**
TemplateCategoryIDs	element	UniqueIDList	optional	ID of the template category. If this tag is missing, then all template categories will be returned.

Response:

TemplateCategoryResponse				
Name	**Type**	**Data Type**	**Use**	**Comments**
Result	element	ResultStatus	required	Result.
TemplateCategories	element	TemplateCategoryList	optional	Collection of template categories.

TemplateCategoryList				
Name	**Type**	**Data Type**	**Use**	**Comments**
TemplateCategory	element	TemplateCategory	optional/multiple	Template category.

TemplateCategory				
Name	**Type**	**Data Type**	**Use**	**Comments**
TemplateCategoryIDs	element	UniqueIDList	required	IDs of the template category
Name	element	Text	required/multiple	Name of the template category.
Order	element	unsignedInt	optional	A low order is before a high order.

Example: Request to get all template categories
[Request]

```
<TemplateCategoryRequest>
</TemplateCategoryRequest>
```

[Response]

```
<TemplateCategoryResponse>
<Result resultStatusFlag="SUCCESS" />
```

```
<TemplateCategories>
        <TemplateCategory>
                <TemplateCategoryIDs>
                        <UniqueID source="TAC">1</UniqueID>
                </TemplateCategoryIDs>
                <Name language="en">Massage</Name>
                <Name language="de">Massagen</Name>
                <Order>2</Order>
        </TemplateCategory>
        <TemplateCategory>
                <TemplateCategoryIDs>
                        <UniqueID source="TAC">2</UniqueID>
                </TemplateCategoryIDs>
                <Name language="en">Baths</Name>
                <Name language="de">Bäder</Name>
                <Order>3</Order>
        </TemplateCategory>
        <TemplateCategory>
                <TemplateCategoryIDs>
                        <UniqueID source="TAC">3</UniqueID>
                </TemplateCategoryIDs>
                <Name language="en">Sport</Name>
                <Name language="de">Sport</Name>
                <Order>1</Order>
        </TemplateCategory>
        <TemplateCategory>
                <TemplateCategoryIDs>
                        <UniqueID source="TAC">4</UniqueID>
                </TemplateCategoryIDs>
                <Name language="en">Golf</Name>
                <Name language="de">Golf</Name>
                <Order>4</Order>
        </TemplateCategory>
        <TemplateCategory>
                <TemplateCategoryIDs>
                        <UniqueID source="TAC">5</UniqueID>
                </TemplateCategoryIDs>
                <Name language="en">Retail</Name>
                <Name language="de">Produkte</Name>
                <Order>5</Order>
        </TemplateCategory>
        <TemplateCategory>
                <TemplateCategoryIDs>
                        <UniqueID source="TAC">7</UniqueID>
```

```
            </TemplateCategoryIDs>
            <Name language="en">Gift certificates</Name>
            <Name language="de">Gutscheine</Name>
            <Order>6</Order>
        </TemplateCategory>
</TemplateCategories>
</TemplateCategoryResponse>
```

40.2 Definition TemplateInfo

(SSF ⇨ SSS)

This request is used to retrieve one or more templates. If no filter is set, all templates will be requested. A template must have one of the following types:

- ACTIVITY
- DINING
- RETAIL
- GIFTCERTIFICATE
- SHIPPINGCOSTS

Soap Action http://htng.org/PWS/2008A/GuestSelfService#TemplateInfo

Request:

TemplateInfoRequest				
Name	**Type**	**Data Type**	**Use**	**Comments**
TemplateIDs	element	UniqueIDList	optional	IDs of the template.
Type	element	string	optional	Type of the template. (possible values: ACTIVITY, DINING, RETAIL, GIFTCERTIFICATE, SHIPPINGCOSTS)
ResortId	element	string	optional	Property code.

Response:
The response delivers all templates, which applies to the given filter criteria.

TemplateInfoResponse				
Name	**Type**	**Data Type**	**Use**	**Comments**
Result	element	ResultStatus	required	Result.
Templates	element	TemplateList	optional	Collection of the found templates.

TemplateList				
Name	**Type**	**Data Type**	**Use**	**Comments**
Template	element	Template	optional/multiple	Template.

Template				
Name	**Type**	**Data Type**	**Use**	**Comments**
type	attribute	string	required	Type of the template. (possible values: ACTIVITY, DINING, RETAIL,

				GIFTCERTIFICATE, SHIPPINGCOSTS)
TemplateIDs	element	UniqueIDList	required	IDs of the template.
Name	element	Text	required/multiple	Name of the template.
TemplateCategories	element	TemplateCategoryList	required	List of template categories. At least one template category must be assigned.
ValidFrom	element	date	optional	Start of the validity period of the template
ValidUntil	element	date	optional	End of the validity period of the template
Description	element	Text	optional/multiple	Detailed description of the template
Note	element	Text	optional/multiple	Additional notes about the template. Can be used for e.g. instructions, ...
Prices	element	PriceList	optional	Contains information about the possible prices.
RelatedItems	element	RelatedItemList	optional	The related items can be used to promote templates or template categories, which are in relation with this template.
Media	element	Media	optional	Media of the template. For example: image, pdf, ...
Extended elements for Activities				
Location	element	string	optional	Location of the template.
AlternativeLocations	element	AlternativeLocationList	optional	List of alternative locations, where this template is also available.
MinimumInAdvanceBookings	element	duration	optional	It is only possible to book this template before this offset, which is calculated from now to the starting point of the booking.
MaximumInAdvanceBookings	element	duration	optional	It is only possible to book this template after this offset, which is calculated from now to the starting point of the booking.
PaymentConditions	element	LinkedPaymentConditionList	optional	Payment conditions of the template. A template can have multiple payment conditions.
AvailableResources	element	ResourceList	optional	All available resources for the template are listed. These resources can be used in requests to define preferred resources.
StartTime	element	string	optional	The booking must be between the Start- and EndTime.
EndTime	element	string	optional	The booking must be between the Start- and EndTime.
StartInterval	element	duration	optional	For example: The Starttime = 08:00, EndTime = 10:00 and the StartInterval = 30 minutes: Possible starts for bookings in this case are: 08:00, 08:30, 09:00, 09:30, 10:00
ActivityType	element	string	optional	ActivityType of the template. (e.g. Massage, Golf, ..)
Duration	element	duration	optional	Duration of the template.
Extended elements for Dining				
Location	element	string	optional	Location of the template.
AlternativeLocations	element	AlternativeLocationList	optional	List of alternative locations, where this template is also available.
MinimumInAdvanceBookings	element	duration	optional	It is only possible to book this template before this offset, which is calculated from now to the starting point of the booking.
MaximumInAdvanceBookings	element	duration	optional	It is only possible to book this template after this offset, which is calculated from now to the starting point of the booking.
PaymentConditions	element	LinkedPaymentConditionList	optional	Payment conditions of the template. A template can have multiple payment conditions.
AvailableResources	element	ResourceList	optional	All available resources for the template are listed. These resources can be used in requests to define preferred resources.

StartTime	element	string	optional	The booking must be between the Start- and EndTime.
EndTime	element	string	optional	The booking must be between the Start- and EndTime.
StartInterval	element	duration	optional	For example: The Starttime = 08:00, EndTime = 10:00 and the StartInterval = 30 minutes: Possible starts for bookings in this case are: 08:00, 08:30, 09:00, 09:30, 10:00
Extended elements for GiftCertificate				
MinimumInAdvanceBookings	element	duration	optional	It is only possible to book this template before this offset, which is calculated from now to the starting point of the booking.
MaximumInAdvanceBookings	element	duration	optional	It is only possible to book this template after this offset, which is calculated from now to the starting point of the booking.
DateSelection	element	unsignedInt	optional	0 … default; 1 … when the gift certificate is selected a date must be selected (check contingents)
GiftCertificateType	element	string	optional	Type of gift certificate: HOTEL,SERVICE,VALUE,...
GiftCertificatePrintName	element	Text	optional/multiple	Additional name, which can be printed on the gift certificate.
GiftCertificatePrintText1	element	Text	optional/multiple	Additional text, which can be printed on the gift certificate.
GiftCertificatePrintText2	element	Text	optional/multiple	Additional text, which can be printed on the gift certificate.
GiftCertificatePrintText3	element	Text	optional/multiple	Additional text, which can be printed on the gift certificate.
Extended elements for Retail				
Weight	element	double	optional	Weight of the retail product, for calculating the shipping costs
Stock	element	Int	optional	Quantity of retail product, which are in stock.

AlternativeLocationList				
Name	**Type**	**Data Type**	**Use**	**Comments**
AlternativeLocation	element	string	optional/multiple	Name of the location.

LinkedPaymentConditionList				
Name	**Type**	**Data Type**	**Use**	**Comments**
PaymentConditionIDs	element	UniqueIDList	optional/multiple	IDs of the payment condition.

PriceList				
Name	**Type**	**Data Type**	**Use**	**Comments**
Price	element	Price	optional/multiple	One possible price of the template.

Price				
Name	**Type**	**Data Type**	**Use**	**Comments**
Name	element	Text	optional/multiple	Name of the price.
Price	element	Amount	optional	Gross price of the template.
Taxes	element	TaxList	optional	Taxes of this price.
Description	element	Text	optional/multiple	Detailed description for this price.

RelatedItemList				
Name	**Type**	**Data Type**	**Use**	**Comments**
RelatedTemplateCategories	element	RelatedTemplateCategoryList	optional	Related template categories.

RelatedTemplates	element	RelatedTemplateList	optional	Related templates.

RelatedTemplateCategoryList				
Name	**Type**	**Data Type**	**Use**	**Comments**
RelatedTemplateCategory	element	RelatedTemplateCategory	optional/multiple	Template category with related item type.

RelatedTemplateCategory				
Name	**Type**	**Data Type**	**Use**	**Comments**
relatedItemType	attribute	string	optional	There are at least 3 possible types - UPSELL, CROSSSELL and ALTERNATE.
TemplateCategoryIDs	element	UniqueIDList	required	IDs of the template category.

RelatedTemplateList				
Name	**Type**	**Data Type**	**Use**	**Comments**
RelatedTemplate	element	RelatedTemplate	optional/multiple	Template with related item type.

RelatedTemplate				
Name	**Type**	**Data Type**	**Use**	**Comments**
relatedItemType	attribute	string	optional	There are at least 3 possible types - UPSELL, CROSSSELL and ALTERNATE.
TemplateIDs	element	UniqueIDList	required	IDs of the template.

TemplateCategoryList				
Name	**Type**	**Data Type**	**Use**	**Comments**
TemplateCategoryIDs	element	UniqueIDList	optional/multiple	List of template category IDs.

Example: Request to get all templates of the property HOTELB
[Request]

```
<TemplateInfoRequest>
        <ResortId>HOTELB</ResortId>
</TemplateInfoRequest>
```

[Response]

```
<TemplateInfoResponse>
<Result resultStatusFlag="SUCCESS" />
<Templates>
        <Template type="ACTIVITY">
                <TemplateIDs>
                        <UniqueID source="TAC">123</UniqueID>
                </TemplateIDs>
                <Name language="en">Full body massage</Name>
                <Name language="de">Ganzkörpermassage</Name>
                <TemplateCategories>
```

```
<TemplateCategoryIDs>
            <UniqueID source="TAC">1</UniqueID>
</TemplateCategoryIDs>
</TemplateCategories>
<Location>Spa</Location>
<AlternativeLocations>
        <AlternativeLocation>Beachfront Spa</AlternativeLocation>
</AlternativeLocations>
<ValidFrom>2001-01-01</ValidFrom>
<ValidUntil>2009-12-31</ValidUntil>
<MinimumInAdvanceBookings>P0Y0M2D</MinimumInAdvanceBookings>
<MaximumInAdvanceBookings>P0Y0M180D</MaximumInAdvanceBookings>
<Description language="en">Massage for the full body.</Description>
<Description language="de">Massage für den ganzen Körper.</Description>
<PaymentConditions>
        <PaymentConditionIDs>
                <UniqueID source="TAC">1</UniqueID>
        </PaymentConditionIDs>
        <PaymentConditionIDs>
                <UniqueID source="TAC">3</UniqueID>
        </PaymentConditionIDs>
</PaymentConditions>
<Prices>
<Price>
                <Name language="en">Full body massage</Name>
                <Name language="de">Ganzkörpermassage</Name>
                <Price currencyCode="USD">130.5</Price>
                <Taxes>
                 <Tax>
                        <TaxName>10 %</TaxName>
                        <TaxRate>10</TaxRate>
                        <TaxAmount>11.86</TaxAmount>
                 </Tax>
                </Taxes>
                <Description language="en">Massage for the full body.</Description>
                <Description language="de">Massage für den ganzen Körper.</Description>
        </Price>
        <Price>
                <Name language="en">Full body massage - Holiday</Name>
                <Name language="de">Ganzkörpermassage - Feiertag</Name>
                <Price currencyCode="USD">150</Price>
                <Taxes>
                 <Tax>
                        <TaxName>10 %</TaxName>
```

```
                        <TaxRate>10</TaxRate>
                        <TaxAmount>13.64</TaxAmount>
                    </Tax>
                    </Taxes>
                    <Description language="en">Massage for the full body.</Description>
                    <Description language="de">Massage für den ganzen Körper.</Description>
            </Price>
    </Prices>
    <AvailableResources>
    <Resource category="Therapists">John Miller</Resource>
    <Resource category="Therapists">Lucy Banner</Resource>
    <Resource category="Therapists">Jack Edwards</Resource>
    <Resource category="Rooms">Massage Room</Resource>
    <Resource category="Rooms">Room 10</Resource>
    <Resource category="Rooms">Room 12</Resource>
    </AvailableResources>
    <RelatedItems>
            <RelatedTemplateCategories>
            <RelatedTemplateCategory relatedItemType="UPSELL">
                    <TemplateCategoryIDs>
                            <UniqueID source="TAC">1</UniqueID>
                    </TemplateCategoryIDs>
            </RelatedTemplateCategory>
            <RelatedTemplateCategory relatedItemType="UPSELL">
                    <TemplateCategoryIDs>
                            <UniqueID source="TAC">2</UniqueID>
                    </TemplateCategoryIDs>
            </RelatedTemplateCategory>
            <RelatedTemplateCategory relatedItemType="UPSELL">
                    <TemplateCategoryIDs>
                            <UniqueID source="TAC">3</UniqueID>
                    </TemplateCategoryIDs>
            </RelatedTemplateCategory>
            <RelatedTemplateCategory relatedItemType="UPSELL">
                    <TemplateCategoryIDs>
                            <UniqueID source="TAC">4</UniqueID>
                    </TemplateCategoryIDs>
            </RelatedTemplateCategory>
    </RelatedTemplateCategories>
            <RelatedTemplates>
            <RelatedTemplate relatedItemType="ALTERNATE">
                    <TemplateIDs>
                            <UniqueID source="TAC">124</UniqueID>
                    </TemplateIDs>
```

```
                </RelatedTemplate>
                <RelatedTemplate relatedItemType="ALTERNATE">
                        <TemplateIDs>
                                <UniqueID source="TAC">125</UniqueID>
                        </TemplateIDs>
                </RelatedTemplate>
                </RelatedTemplates>
        </RelatedItems>
        <ActivityType>Spa Massage</ActivityType>
        <Duration>P0Y0M0DT0H60M</Duration>
        <StartTime>08:00</StartTime>
        <EndTime>10:00</EndTime>
        <StartInterval>P0Y0M0DT0H30M</StartInterval>
        <Media>
                <MediaIDs>
                        <UniqueID source="TAC">950</UniqueID>
                </MediaIDs>
        <MediaFormat>image/jpeg</MediaFormat>
                <MediaURI>\\192.168.22.70\data\itempictures\0\0\0\950.jpg</MediaURI>
                <MediaData>/9j/4AAQSkZJRgABAgAAZABkAAD/7AARRHVja3kAAQAEAAAAPAAA/+4ADkFkb2JlAGTAAAAAAf/b
                AIQABgQEBAUEBgUFBgkGBQYJCwgGBggLDAoKCwoKDBAMDAwMDAwQDA4PEA8ODBMTFBQTExwbGxsc
                Hx8fHx8fHx8fHwEHBwcNDA0YEBAYGhURFRofHx8fHx8fHx8fHx8fHx8fHx8fHx8fHx8fHx8f
                Hx8fHx8fHx8fHx8fHx8fHx8f/8AAEQgAMgAyAwERAAIRAQMRAf/EAHwAAQABBQEBAAAAAAAAAAA
                AAAIAQQFBgcDCQEBAQEBAQEAAAAAAAAAAAAAAAECAwQFEAACAgECBQMDAwUAAAAAAAABAgMEABEF
                IRITBgcxYSJBcZFRwRSBMpIVCBEBAQACAQQDAQAAAAAAAAAAAAERAgMxQXESIZEiBP/aAAwDAQAC
                EQMRAD8AITgMCjuqKXchUUEsxOgAHqScDG1e5+2rdkVau7U7FknQQRWIncn2VWJwuKyeEMBgMBga
                L5zleLxJ3RIjOrLTb5RnRgOZdeI9vXDWnVCLb90uvum3xLKVc264SSMaSgmZR8WHHXLh3fRfI8xg
                MBgeF67VoUrF62/Tq1Y3mnkIJ5Y41LMdBqeAGBwTyL5yO9bTd2bYavTp3Y2rz2bK6u8UgKuFj9F1
                U/XLhMuQ7VS2ypuFWz01pvVnisxz14I5XEsLh0/vK6DVfp+MF3qR3ZHnLZt4v1tn3Rkh3Gy6w1po
                wwSSRuCqyMOZGJ+4+2MJK6lkaMBga35Jm6XYHcLa8f8AX2B+YyP3wIghbU1t0Rm6cZC8qBddT7sD
                mpHz+f8Ao212+FlU3rb5mjDTyo05YQIyxhpOTgxQKGPqNOIBxhdtuads+GV8f3o7HkHYzE0jLBul
                LmMqdM6tMpHAgH0/UZLHXhu8v77ptZHqMCz3S09aDnX85KsaD5C3iaXsbfo2IIkqMn+TKP3ySlxh
                Faxtz296hk61eGKOaNyZnZGPTmLSLy8jB9UHDiPfOseS8musuetyuKHbsMCU0S1CGpx2oJHV44lk
                jtEj1+cqlPoeXK47c8ufMv0zXYm00qPduytVlWcC7SVp0cuGMcyaDXRR8fYZNnXi2u99qmgp1GYe
                tXAo6K45WGoP0OBrXdPZVXedpuUY36H8uMxsQNQNeIOnsRkwOJ7h/wA7d09Yywy1LDD0fmkhY/0A
                YZc1MLdfA3fA0H8esoHDUTken2TLln0jYez/AAhvtDfKFzcJIErVZ452hiaR2YxNzKOZgoHyHHJl
                Zq7wo0UDDSuAwGAwGAwGAwP/2QA=</MediaData>
        </Media>
        <ResortId>HOTELB</ResortId>
    </Template>
</Templates>
</TemplateInfoResponse>
```

40.3 Definition PaymentCondition

(SSF ⇨ SSS)

This request is used to retrieve one or more payment conditions. The following filter can be set to request payment conditions:

Soap Action http://htng.org/PWS/2008A/GuestSelfService#PaymentCondition

Request:

PaymentConditionRequest				
Name	**Type**	**Data Type**	**Use**	**Comments**
PaymentConditionIDs	element	UniqueIDList	optional	IDs of the payment condition.
ResortId	element	string	optional	Property code.

Response:
The PaymentConditionResponse delivers a collection of payment conditions.

PaymentConditionResponse				
Name	**Type**	**Data Type**	**Use**	**Comments**
Result	element	ResultStatus	required	Lookup result. If successful, the responding system is only required to set the resultStatusFlag as SUCCESS, and return the data in the PaymentConditionList.
PaymentConditions	element	PaymentConditionList	optional	Collection of payment conditions.

PaymentConditionList				
Name	**Type**	**Data Type**	**Use**	**Comments**
PaymentCondition	element	PaymentCondition	optional/multiple	Payment condition.

PaymentCondition				
Name	**Type**	**Data Type**	**Use**	**Comments**
type	attribute	string	required	The attribute type is required and the following values are available: PREPAYMENT, FULLPAYMENT, BALANCEPAYMENT, CANCELLATION
PaymentConditionIDs	element	UniqueIDList	required	IDs of the payment condition.
Value	element	PaymentConditionValue	required	Value of the payment condition.
CancellationWindow	element	duration	optional	Time offset, which are only used in case of cancellations. If no offset is given, the cancellation condition applies every time.
DueDate	element	duration	optional	Due date of the payment.
Taxes	element	TaxList	optional	Taxes of the payment condition.
Description	element	Text	optional/multiple	Detailed description of the payment condition.
ResortId	element	string	required	Property code.

PaymentConditionValue

Name	Type	Data Type	Use	Comments
type	attribute	string	required	Either PERCENTAGE or FIXPRICE.
Value	element	Amount	required	Value of the payment condition.

Example: Request to get all payment conditions of property HOTELB
[Request]

```
<PaymentConditionRequest>
        <ResortId>HOTELB</ResortId>
</PaymentConditionRequest>
```

[Response]

```
<PaymentConditionResponse>
<Result resultStatusFlag="SUCCESS" />
<PaymentConditions>
        <PaymentCondition type="PREPAYMENT">
                <PaymentConditionIDs>
                        <UniqueID source="TAC">1</UniqueID>
                </PaymentConditionIDs>c
                <Value type="PERCENTAGE">
                <Value>10</Value>
                </Value>
                <Description language="en">10% prepayment</Description>
                <Description language="de">10% Anzahlung</Description>
                <ResortId>HOTELB</ResortId>
        </PaymentCondition>
        <PaymentCondition type="PREPAYMENT">
                <PaymentConditionIDs>
                        <UniqueID source="TAC">2</UniqueID>
                </PaymentConditionIDs>
                <Value type="FIXPRICE">
                <Value currencyCode="USD">15</Value>
                </Value>
                <Description language="en">15$ prepayment</Description>
                <Description language="de">15$ Anzahlung</Description>
                <ResortId>HOTELB</ResortId>
        </PaymentCondition>
        <PaymentCondition type="CANCELLATION">
                <PaymentConditionIDs>
                        <UniqueID source="TAC">3</UniqueID>
                </PaymentConditionIDs>
                <Value type="PERCENTAGE">
                <Value>50</Value>
                </Value>
```

```
            <CancellationWindow>P0Y0M0DT48H</CancellationWindow>
            <Description language="en">50% cancellation costs within two days</Description>
            <Description language="de">50% Stornogebühr innerhalb von zwei Tagen</Description>
            <ResortId>HOTELB</ResortId>
        </PaymentCondition>
        <PaymentCondition type="CANCELLATION">
            <PaymentConditionIDs>
                <UniqueID source="TAC">4</UniqueID>
            </PaymentConditionIDs>
            <Value type="PERCENTAGE">
            <Value>100</Value>
            </Value>
            <CancellationWindow>P0Y0M0DT24H</CancellationWindow>
            <Description language="en">100% cancellation costs within one day</Description>
            <Description language="de">100% Stornogebühr innerhalb von einem Tag</Description>
            <ResortId>HOTELB</ResortId>
        </PaymentCondition>
</PaymentCondtions>
</PaymentConditionResponse>
```

40.4 Definition Availability

(SSF ⇨ SSS)

The selected availabilities will be finalized(created) by a Create request. An availability request is only necessary for activity and dining templates (not for retail and gift certificates).

Possible scenario (dependent on the implementation of the SSS):
Each returned availability can be reserved by a temporary booking in the SSS to reserve the inventory, therapists, etc. The remaining temporary bookings will be deleted automatically.

Soap Action http://htng.org/PWS/2008A/GuestSelfService#Availability

Request

AvailabilityRequest				
Name	**Type**	**Data Type**	**Use**	**Comments**
availabilityType	attribute	string	optional	BESTPRICE or BESTTIME.
TemplateIDs	element	UniqueIDList	required	IDs of the template.
ModifyBookingIDs	element	UniqueIDList	optional	Used to modify an existing booking in the SSS. The SSS has to know the old booking ID to exclude them at the availability check.
Profile	element	Profile	required	Complete profile, because guest type, membership type, etc. could be affect the availability and the price.
NumberOfPersons	element	unsignedInt	optional	Number of persons. Default value = 1. This field is needed for the template type dining. (For example: Dinner for 10 persons)
TimeSpans	element	TimeSpanList	required	This element contains the possible time spans.

PreferredResources	element	ResourceList	optional	This resources will be preferred if there are possible results with them. The PreferredResources can contain many Resource entries.
Preferences	element	PreferenceElementList	optional	Preference list.
ResortId	element	string	required	Property code.

Response

AvailabilityResponse				
Name	**Type**	**Data Type**	**Use**	**Comments**
Result	element	ResultStatus	required	Availabilityresult. If successful, the responding system is only required to set the resultStatusFlag as SUCCESS, and return the data in the AvailabilityList.
Availabilities	element	AvailabilityList	optional	All found availabilities. The maximum quantity of returned availabilities should be configurable in the SSS.

AvailabilityList				
Name	**Type**	**Data Type**	**Use**	**Comments**
Availability	element	Availability	optional/multiple	Availability.

Availability				
Name	**Type**	**Data Type**	**Use**	**Comments**
availabilityType	attribute	string	optional	BESTPRICE or BESTTIME.
TemplateIDs	element	UniqueIDList	required	IDs of the template.
BookingIDs	element	UniqueIDList	required	IDs of the temporary booking. Must be used to create this booking if would be selected.
TimeSpan	element	OGTimeSpan	required	Time span of the booking.
Price	element	Amount	optional	Price of the booking.
Resources	element	ResourceList	optional	Resources of the booking.
ResortId	element	string	required	Property code.

Example:
[Request]

```
<AvailabilityRequest availabilityType="BESTPRICE">
        <TemplateIDs>
              <UniqueID source="TAC">123</UniqueID>
        </TemplateIDs>
        <Profile>
        <IDs>
                <UniqueID source="ES">96</UniqueID>
                <UniqueID source="TAC">10968</UniqueID>
        </IDs>
        <Customer>
                ........
        </Customer>
        </Profile>
        <TimeSpans>
                <TimeSpan>
                        <Start>2007-05-15T08:30:00Z</Start>
                        <End>2007-05-15T09:30:00Z</End>
```

```
                </TimeSpan>
                <TimeSpan>
                        <Start>2007-05-15T10:00:00Z</Start>
                        <End>2007-05-15T12:00:00Z</End>
                </TimeSpan>
        </TimeSpans>
        <PreferredResources>
                <Resource category="Therapists">John Miller</Resource>
        </PreferredResources>
        <Preferences>
        < Preference preferenceType="GENDER" preferred="MALE" mode="PREFERRED">Therapists</Preference>
        </Preferences>
        <ResortId>HOTELB</ResortId>
</AvailabilityRequest>
```

[Response]

```
<AvailabilityResponse>
<Result resultStatusFlag="SUCCESS" />
  <Availabilities>
        <Availability availabilityType="BESTPRICE">
                <TemplateIDs>
                    <UniqueID source="TAC">123</UniqueID>
                </TemplateIDs>
                <BookingIDs>
                        <UniqueID source="TAC">67888</UniqueID>
                </BookingIDs>
                <TimeSpan>
                        <Start>2007-05-15T08:30:00Z</Start>
                        <End>2007-05-15T09:30:00Z</End>
                </TimeSpan>
                <Price currencyCode="USD">130.5</Price>
                <Resources>
                        <Resource category="Therapists">John Miller</Resource>
                <Resource category="Rooms">Room 10</Resource>
                </Resources>
                <ResortId>HOTELB</ResortId>
        </Availability>
  </Availabilities>
</AvailabilityResponse>
```

40.5 Definition Create

(SSF ⇨ SSS)

The Create request can contain many Booking elements and one PaymentGuarantee element.

This message is the confirmation for a booking which was returned by the Availability request. With this request, the availability of the availability request, will be finalized in the SSS using the following information:

Soap Action http://htng.org/PWS/2008A/GuestSelfService#Create

Request:

CreateRequest				
Name	**Type**	**Data Type**	**Use**	**Comments**
Bookings	element	BookingIdentificationList	required	Collection of bookings.
ModifyBookingIDs	element	UniqueIDList	optional	Used to modify an existing booking in the SSS. The SSS has to know the old booking ID to delete them.
PaymentGuarantee	element	PaymentIdentification	optional	It should be configurable, if a payment guarantee is necessary to create a booking.

Response:
The CreateResponse contains all requested bookings with their creation result.

CreateResponse				
Name	**Type**	**Data Type**	**Use**	**Comments**
Bookings	element	BookingResultList	required	Collection of booking results.

Example:

[Request]

```
<CreateRequest>
        <Bookings>
                <Booking>
                        <BookingIDs>
                                <UniqueID source="TAC">67888</UniqueID>
                                <UniqueID source="ES">150</UniqueID>
                        </BookingIDs>
                        <ReferenceNumber>235667</ReferenceNumber>
                </Booking>
        </Bookings>
        <PaymentGuarantee>
        <CreditCard>
                <CardNumber>4444333322221111</CardNumber>
                <ExpireDate>1008</ExpireDate>
        </CreditCard>
        </PaymentGuarantee>
</CreateRequest>
```

[Response]

```
<CreateResponse>
 <Bookings>
        <Booking>
                <Result resultStatusFlag="SUCCESS" />
                <BookingIDs>
                        <UniqueID source="TAC">67888</UniqueID>
                        <UniqueID source="ES">150</UniqueID>
                </BookingIDs>
        </Booking>
  </Bookings>
</CreateResponse>
```

40.6 Definition Delete

(SSF ⇨ SSS)

This request deletes bookings which were created by the Create request.

Soap Action http://htng.org/PWS/2008A/GuestSelfService#Delete

Request:

DeleteRequest				
Name	**Type**	**Data Type**	**Use**	**Comments**
Bookings	element	BookingIdentificationList	required	Collection of bookings, which should be deleted.

Response:

DeleteResponse				
Name	**Type**	**Data Type**	**Use**	**Comments**
Bookings	element	BookingResultList	required	Collection of bookings, with their result.

Example:
[Request]

```
<DeleteRequest>
        <Bookings>
                <Booking>
                        <BookingIDs>
                                <UniqueID source="TAC">67888</UniqueID>
                                <UniqueID source="ES">150</UniqueID>
                        </BookingIDs>
                </Booking>
```

```
        </Bookings>
</DeleteRequest>
```

[Response]

```
<DeleteResponse>
        <Bookings>
                <Booking>
                        <BookingIDs>
                                <UniqueID source="TAC">67888</UniqueID>
                                <UniqueID source="ES">150</UniqueID>
                        </BookingIDs>
                        <Result resultStatusFlag="SUCCESS" />
                </Booking>
        </Bookings>
</DeleteResponse>
```

40.7 Definition Sale

(SSF ⇨ SSS)

The Sale request contains all necessary information for completion a sales process in the SSS.

Soap Action http://htng.org/PWS/2008A/GuestSelfService#Sale

Request:

A Sale request consists of the following parts:

- Profile
- Payments (arbitrary number of payments)
- Items (arbitrary number of Booking, Retail, GiftCertificate and ShippingCosts elements)

The element Profile corresponds to the structure of a New/UpdateProfile from the guest synchronisation.

SaleRequest				
Name	**Type**	**Data Type**	**Use**	**Comments**
ProfileIdentification	element	ProfileIdentification	required	Used to identify the profile.
Payments	element	PaymentList	required	Collection of payments.
Items	element	ItemList	required	Collection of items.
ResortId	element	string	required	Property code.

ProfileIdentification				
Name	**Type**	**Data Type**	**Use**	**Comments**
Profile	element	Profile	optional/choice	Profile as defined in SGI specification. (Choice)
ProfileIDs	element	UniqueIDList	optional/choice	ProfileIDs. (Choice)

PaymentList				
Name	**Type**	**Data Type**	**Use**	**Comments**
Payment	element	Payment	required/multiple	The element Payment must be set for every used payment method in the selling process.

Payment				
Name	**Type**	**Data Type**	**Use**	**Comments**
method	attribute	string	required	Sets the payment method in the SSS.
Value	element	Amount	required	With this payment method paid value.
PaymentInformation	element	PaymentIdentification	optional	When the guest pays with a gift certificate, credit card or room the additional payment information must be set here.

ItemList				
Name	**Type**	**Data Type**	**Use**	**Comments**
Booking	element	BookingItem	optional/multiple	This booking has to be created by a prior Create message.
CreateBooking	element	BookingIdentification	optional/multiple	This element has the same function like the single message CreateRequest. This element should be used to create and sell bookings in one step without a prior Create message.
GiftCertificate	element	GiftCertificateItem	optional/multiple	The created gift certificate numbers will be returned by the SaleResponse.
Item	element	Item	optional/multiple	Common sales item.
ShippingCosts	element	ShippingCostItem	optional/multiple	The ShippingCosts must be calculated by the weight of the ordered retail products.

BookingItem				
Name	**Type**	**Data Type**	**Use**	**Comments**
BookingIDs	element	UniqueIDList	required	IDs of the booking.
Type	element	string	optional	Full or partial sell of an existing booking or cancellation of an booking and book the cancellation costs (Type = CANCEL)) Possible values: PREPAYMENT, FULLPAYMENT, BALANCEPAYMENT, CANCELLATION Default: FULLPAYMENT
Price	element	Amount	required	Price of the booking.

GiftCertificateItem				
Name	**Type**	**Data Type**	**Use**	**Comments**
TemplateIDs	element	UniqueIDList	required	IDs of the template.
Price	element	Amount	required	Amount of the gift certificate.
BuyerName	element	string	optional	Name of the buyer (donor).
DedicationText	element	string	optional	The content of this field should be changeable for the buyer of the gift

				certificate, because this text could be printed on the gift certificate as dedication for the recipient.
LanguageCode	element	string	required	Needed to choose the right text blocks for the gift certificate. Possible values: en,de,fr,...

Item				
Name	**Type**	**Data Type**	**Use**	**Comments**
TemplateIDs	element	UniqueIDList	required	IDs of the template.
Price	element	Amount	required	Price of the sales item. (e.g. retail product)

ShippingCostItem				
Name	**Type**	**Data Type**	**Use**	**Comments**
Price	element	Amount	required	Price of the shipping costs.
Description	element	Text	optional	This text could be used to present the type of shipping(e.g. overnight express) on the bill.
TemplateID	element	UniqueID	optional	TemplateID of shipping cost template.

Response:

SaleResponse				
Name	**Type**	**Data Type**	**Use**	**Comments**
Result	element	ResultStatus	required	Result.
InvoiceNumber	element	string	required	Number of the created invoice.
InvoiceDate	element	date	required	Date and time of the created invoice.
ReceiptImage	element	Media	optional	Receipt media.
GiftCertificates	element	GiftCertificateList	optional	This element is optional, because it is possible that the sale request contains no gift certificate item.

GiftCertificateList				
Name	**Type**	**Data Type**	**Use**	**Comments**
GiftCertificate	element	GiftCertificateData	optional/multiple	Collection of the created gift certificates.

GiftCertificateData				
Name	**Type**	**Data Type**	**Use**	**Comments**
GiftCertificateNumber	element	string	required	Number of the created gift certificate.
GiftCertificateData	element	Media	optional	Depending on the workflow, this element is needed. (e.g. to create the gift certificate layout print in the SSS and send the base64 coded image or pdf to the SSF)

Example:
[Request]

```
<SaleRequest>
        <Profile nameType="GUEST" languageCode="en" nationality="AT"
        <IDs>
```

```
        <UniqueID source="ES">96</UniqueID>
        <UniqueID source="TAC">10968</UniqueID>
</IDs>
<Customer gender="MALE">
<PersonName familiarName="Rappold">
        <NameTitle />
        <FirstName>Bernhard</FirstName>
        <LastName>Rappold</LastName>
</PersonName>
</Customer>
<Addresses>
        <NameAddress addressType="HOME" primary="true">
                <AddressLine>Schildbach 111</AddressLine>
                <CityName />
                <StateProv />
                <CountryCode>AT</CountryCode>
                <PostalCode />
        </NameAddress>
        </Addresses>
        <Phones>
                <NamePhone phoneType="EMAIL" phoneRole="EMAIL" primary="true">
                <PhoneNumber>b.rappold@reservationassistant.com</PhoneNumber>
                </NamePhone>
        </Phones>
</Profile>
<Payments>
<Payment method="visa">
                <Value currencyCode="USD">130.5</Value>
                <PaymentInformation>
                <CreditCard>
                        <CardNumber>4444333322221111</CardNumber>
                        <ExpireDate>1008</ExpireDate>
                        <CVV2>999</CVV2>
                        <CardHolderName>RAPPOLD BERNHARD</CardHolderName>
                        <CardHolderZipCode></CardHolderZipCode>
                        <CardHolderAddress></CardHolderAddress>
                        <CardHolderCity></CardHolderCity>
                        <RemoteIP>81.223.207.12</RemoteIP>
                </CreditCard>
                </PaymentInformation>
        </Payment>
</Payments>
<Items>
        <Booking>
```

```
                <BookingIDs>
                        <UniqueID source="TAC">67888</UniqueID>
                        <UniqueID source="ES">150</UniqueID>
                </BookingIDs>
                <Price currencyCode="USD">130.50</Price>
            </Booking>
        </Items>
        <ResortId>HOTELB</ResortId>
</SaleRequest>
```

[Response]

```
<SaleResponse>
        <Result resultStatusFlag="SUCCESS" />
        <InvoiceNumber>10123</InvoiceNumber>
        <InvoiceDate>2007-05-14</InvoiceDate>
</SaleResponse>
```

40.8 Definition FetchBooking

(SSF ⇨ SSS)

With this request the SSF can retrieve detailed information for bookings. (e.g. the trainer, room, .. of an activity, …).

Soap Action http://htng.org/PWS/2008A/GuestSelfService#FetchBooking

Request:
One of the filter must be set.

FetchBookingRequest				
Name	**Type**	**Data Type**	**Use**	**Comments**
BookingIDs	element	UniqueIDList	optional	To request one specific booking.
ReferenceNumber	element	string	optional	To request all bookings for a global reference number.
Type	element	string	optional	To request bookings with a specific template type.
TimeSpan	element	OGTimeSpan	optional	To reuqest all bookings between this time span.
ProfileIDs	element	UniqueIDList	optional	To request all bookings for a profile.
PersonName	element	PersonName	optional	To request bookings by guest name.
Membership	element	NameMembership	optional	To request all bookings for a membership. (e.g. MembershipNumber)
EmailAddress	element	string	optional	To request bookings by email address.
Reservation	element	ReservationIdentification	optional	To request all bookings by one hotel reservation.

Response:
The response can contain a lot of Activity and Dining elements.

FetchBookingResponse				
Name	**Type**	**Data Type**	**Use**	**Comments**
Result	element	ResultStatus	required	Result.
Bookings	element	BookingList	optional	Collection of the found bookings.

BookingList				
Name	**Type**	**Data Type**	**Use**	**Comments**
Booking	element	Booking	optional/multiple	Collection of the found bookings.

Booking				
Name	**Type**	**Data Type**	**Use**	**Comments**
type	attribute	string	required	Type of booked template.
BookingIDs	element	UniqueIDList	required	BookingIDs of the booking. (SGI ActivityID)
ProfileIDs	element	UniqueIDList	required	ProfileIDs of the booking guest.
ActivityType	element	string	optional	Type of the booking. (SGI)
Location	element	string	optional	Location of the booking. (SGI)
Name	element	Text	required/multiple	Name of booking. (SGI)
NumberOfPersons	element	unsignedInt	optional	Number of persons. (Only used for dining) (SGI)
TimeSpan	element	OGTimeSpan	required	Time span of the booking. (SGI)
Duration	element	OGTimeSpan	optional	The length of the booking. (SGI)
Description	element	Text	optional/multiple	Detailed booking description. (SGI)
Note	element	Text	optional/multiple	Additional notes about the booking. (SGI) Can be used for e.g. instructions, ...
Remark	element	string	optional	Remark of the booking. Can be stored to the booking by the Create request.
Price	element	Amount	required	Booking price. (SGI Amount)
Participants	element	ParticipantList	optional	List of booked participants. (SGI Participants)
Resources	element	ResourceList	optional	List of booked resources.
RequestedPreferences	element	PreferenceElementList	optional	Requested preferences of the availability request.
RequestedPreferredResources	element	ResourceList	optional	Requested preferred resources of the availability request.

ParticipantList				
Name	**Type**	**Data Type**	**Use**	**Comments**
Participant	element	PersonName	optional/multiple	Name of the participants.

Example:
[Request]

```
<FetchBookingRequest>
        <BookingIDs>
                <UniqueID source="TAC">67888</UniqueID>
        </BookingIDs>
        <ProfileIDs>
                <UniqueID source="ES">96</UniqueID>
                <UniqueID source="TAC">10968</UniqueID>
```

```
        </ProfileIDs>
        <Type>ACTIVITY</Type>
        <ReferenceNumber>235667</ReferenceNumber>
</FetchBookingRequest>
```

[Response]

```
<FetchBookingResponse>
<Result resultStatusFlag="SUCCESS">
        <Text>
                <TextElement language="en" />
        </Text>
        <IDs>
                <UniqueID source="TAC">67888</UniqueID>
        </IDs>
</Result>
<Bookings>
        <Booking type="ACTIVITY">
        <BookingIDs>
                <UniqueID source="TAC">67888</UniqueID>
        </BookingIDs>
        <ProfileIDs>
                <UniqueID source="ES">96</UniqueID>
                <UniqueID source="TAC">10968</UniqueID>
        </ProfileIDs>
        <ActivityType>Spa Massage</ActivityType>
        <Location>Spa</Location>
        <Name language="en">Full body massage</Name>
        <Name language="de">Ganzkörpermassage</Name>
        <TimeSpan>
                <Start>2007-05-15T08:30:00Z</Start>
                <End>2007-05-15T09:30:00Z</End>
        </TimeSpan>
        <Description language="en">Massage for the full body.</Description>
        <Description language="de">Massage für den ganzen Körper.</Description>
        <Price currencyCode="USD">130.50</Price>
        <Participants>
                <Participant>
                        <NameTitle />
                        <FirstName>Bernhard</FirstName>
                        <LastName>Rappold</LastName>
                </Participant>
        </Participants>
        <Resources>
                <Resource category="Therapists">John Miller</Resource>
```

```
            <Resource category="Rooms">Room 10</Resource>
        </Resources>
        </Booking>
</Bookings>
</FetchBookingResponse>
```

40.9 Definition FetchGiftCertificate

(SSF ⇨ SSS)

With this request the SSF can retrieve detailed information for a gift certificate. (e.g. value, expiration date, ...)
For example, in case of gift certificate payment the SSF can request the value of the gift certificate by this request.

Soap Action http://htng.org/PWS/2008A/GuestSelfService#FetchGiftCertificate

Request:

FetchGiftCertificateRequest				
Name	**Type**	**Data Type**	**Use**	**Comments**
GiftCertificate	element	GiftCertificateIdentification	required	To request one specific gift certificate.

Response:

FetchGiftCertificateResponse				
Name	**Type**	**Data Type**	**Use**	**Comments**
Result	element	ResultStatus	required	Result.
GiftCertificate	element	GiftCertificate	optional	Found gift certificate.

GiftCertificate				
Name	**Type**	**Data Type**	**Use**	**Comments**
GiftCertificateIdentification	element	GiftCertificateIdentification	required	Identification of the gift certificate.
GiftCertificateType	element	string	required	Type of gift certificate: HOTEL,SERVICE,VALUE,...
GiftCertificateStatus	element	string	optional	Status of gift certificate: PRINTED, PAID, REDEEMED, CANCELLED, ...
TemplateIDs	element	UniqueIDList	optional	Ids of the corresponding template.
LanguageCode	element	string	optional	Language of the gift certificate.
TemplateName	element	string	optional	Name of the corresponding template in gift certificate language.
TemplateDescription	element	string	optional	Description of the corresponding template in gift certificate language.
GiftCertificatePrintName	element	string	optional	Additional name, which can be printed on the gift certificate.
GiftCertificatePrintText1	element	string	optional	Additional text, which can be printed on the gift certificate.
GiftCertificatePrintText2	element	string	optional	Additional text, which can be printed on the gift certificate.
GiftCertificatePrintText3	element	string	optional	Additional text, which can be printed on the gift certificate.
IssueDate	element	date	optional	Date of issue.
ValidFrom	element	date	optional	Valid from.
ValidUntil	element	date	optional	Valid until.
RedeemDate	element	date	optional	Redeem date.

PaidDate	element	date	optional	Paid date.
BuyerProfileIDs	element	UniqueIDList	optional	Buyer ProfileIDs.
BuyerName	element	string	optional	Buyer name (donor). This is no PersonName, because this text can be changed on creation by the customer.
DedicationText	element	string	optional	Dedication text.
Tax	element	Tax	optional	Tax of the gift certificate.
Value	element	Amount	optional	Gift certificate value.
ServiceTemplateIDs	element	UniqueIDList	optional	TemplateIDs of the corresponding service.
InvoiceNumber	element	string	optional	Number of the sale invoice.

Example:
[Request]

```
<FetchGiftCertificateRequest>
        <GiftCertificate>
                <GiftCertificateNumber>VV2300021/07</GiftCertificateNumber>
        <ExpirationDate>2037-01-14</ExpirationDate>
        <VerificationCode>ABC2356671</VerificationCode>
        </GiftCertificate>
</FetchGiftCertificateRequest>
```

[Response]

```
<FetchGiftCertificateResponse>
        <Result resultStatusFlag="SUCCESS"/ >
        <GiftCertificate>
                <GiftCertificateIdentification>
                        <GiftCertificateNumber>VV2300021/07</GiftCertificateNumber>
                <ExpirationDate>2037-01-14</ExpirationDate>
                <VerificationCode>ABC2356671</VerificationCode>
                </GiftCertificateIdentification>
                <GiftCertificateType>VALUE</GiftCertificateType>
                <GiftCertificateStatus>PAID</GiftCertificateStatus>
                <TemplateIDs>
                        <UniqueID source="TAC">675</UniqueID>
                </TemplateIDs>
                <LanguageCode>en</LanguageCode>
                <TemplateName>Value gift certificate 100 USD</TemplateName>
                <TemplateDescription>Value gift certificate 100 USD</TemplateDescription>
                <GiftCertificatePrintName>100 USD value gift certificate</GiftCertificatePrintName>
                <GiftCertificatePrintText1>Prizes cannot be redeemed for money</GiftCertificatePrintText1>
                <IssueDate>2007-01-14</IssueDate>
                <ValidFrom>2007-01-14</ValidFrom>
                <ValidUntil>2037-01-14</ValidUntil>
                <PaidDate>2007-01-14</PaidDate>
                <BuyerProfileIDs>
                <UniqueID source="ES">96</UniqueID>
```

```
            <UniqueID source="TAC">10968</UniqueID>
        </BuyerProfileIDs>
        <BuyerName>Rappold Bernhard</BuyerName>
        <DedicationText>Happy Birthday</DedicationText>
        <Value currencyCode="USD">100.00</Value>
        <InvoiceNumber>635424</InvoiceNumber>
        </GiftCertificate>
</FetchGiftCertificateResponse>
```

Chapter 41 Global schema elements

Each schema element, which is used in more than one message, is defined in this section. The usages of this schema elements are marked with green color. The blue elements are defined in the Single Guest Itinerary specification.

TimeSpanList				
Name	**Type**	**Data Type**	**Use**	**Comments**
TimeSpan	element	OGTimeSpan	required/multiple	One time span is required.

Media				
Name	**Type**	**Data Type**	**Use**	**Comments**
MediaID	element	UniqueIDList	optional	IDs of the media.
MediaFormat	element	string	optional	MIME type of the media. For example: image/jpeg, application/pdf, ..
MediaURI	element	anyURI	optional	URI of the media file (e.g. \\192.168.22.70\data\itempictures\0\0\0\950.jpg) (Choice)
MediaData	element	base64Binary	optional	Base64 coded media data. (Choice)

ResourceList				
Name	**Type**	**Data Type**	**Use**	**Comments**
Resource	element	Resource	optional/multiple	One of the possible resource.

Resource				
Name	**Type**	**Data Type**	**Use**	**Comments**
Resource	extension	string	required	Name of the resource.
category	attribute	string	optional	Name of the resource category.

BookingIdentificationList				
Name	**Type**	**Data Type**	**Use**	**Comments**
Booking	element	BookingIdentification	required/multiple	At least one booking is needed.

BookingIdentification				
Name	**Type**	**Data Type**	**Use**	**Comments**
BookingIDs	element	UniqueIDList	required	The internal BookingID, which was returned by a previous AvailabilityResponse must be filled. The external BookingID can be assigned.
ReferenceNumber	element	string	optional	The reference number is a global number to identify a group of activities, hotel reservations, etc.
Remark	element	string	optional	This remark can be used to store a note to a booking.

BookingResultList				
Name	**Type**	**Data Type**	**Use**	**Comments**
Booking	element	BookingResult	required/multiple	At least one booking result is required.

BookingResult				
Name	**Type**	**Data Type**	**Use**	**Comments**
Result	element	ResultStatus	required	Result of the booking creation.
BookingIDs	element	UniqueIDList	required	IDs of the created bookings.

TaxList				
Name	**Type**	**Data Type**	**Use**	**Comments**
Tax	element	Tax	optional/multiple	List of taxes.

Tax				
Name	**Type**	**Data Type**	**Use**	**Comments**
TaxName	element	Text	optional	Name of tax.
TaxRate	element	double	required	Tax rate. (e.g. 7)
TaxAmount	element	Amount	optional	Tax amount.

PaymentIdentification				
Name	**Type**	**Data Type**	**Use**	**Comments**
CreditCard	element	CreditCardIdentification	optional/choice	The CardNumber and ExpireDate are required in the element CreditCardIdentification.
Reservation	element	ReservationIdentification	optional/choice	Used to identify a hotel reservation.
GiftCertificate	element	GiftCertificateIdentification	optional/choice	Gift certificate payment information.

GiftCertificateIdentification				
Name	**Type**	**Data Type**	**Use**	**Comments**
GiftCertificateNumber	element	string	required	Unique gift certificate number.
ExpirationDate	element	date	optional	Expiration date of gift certificate.
VerificationCode	element	string	optional	Code which is printed on the gift certificate like the CVV2 code of a credit card.

ReservationIdentification				
Name	**Type**	**Data Type**	**Use**	**Comments**
ReservationID	element	UniqueID	optional	Reservation ID.
RoomNumber	element	string	optional	Room number.
ConfirmationNo	element	string	optional	Confirmation number.

CreditCardIdentification				
Name	**Type**	**Data Type**	**Use**	**Comments**
CardNumber	element	string	required	Complete credit card number without spaces.
ExpirationDate	element	string	required	Expiration date of the credit card without separator (e.g. October 2008 is formatted as 1008)
CVV2	element	string	optional	Code for verifying the card.
CardHolderName	element	string	optional	Name of the card holder.
CardHolderAddressLine	element	string	optional	Address line of the card holder.

CardHolderCityName	element	string	optional	City name of the card holder.
CardHolderStateProv	element	string	optional	State of the card holder.
CardHolderCountryCode	element	string	optional	Country of the card holder.
CardHolderPostalCode	element	string	optional	Postal code of the card holder.
RemoteIP	element	string	optional	IP of the remote system. Can be used for logging purpose.

PreferenceElementList				
Name	**Type**	**Data Type**	**Use**	**Comments**
Preference	element	PreferenceElement	optional/multiple	The Preferences can contain many Category entries which must be unique.

PreferenceElement				
Name	**Type**	**Data Type**	**Use**	**Comments**
Category	extension	string	required	Name of the resource category.
preferenceType	attribute	string	required	Type of preference: e.g. GENDER, SMOKING, MEALTYPE, TABLETYPE
preferred	attribute	string	required	Requesting value: GENDER: MALE or FEMALE SMOKING: SMOKER or NONSMOKER ....
mode	attribute	string	required	Mode of the request: PREFERRED or STRICT