Oracle® Hospitality PMS Interface
HGBU-IFC8-XML_POS Interface Specification
Release 2.16
Hotel Property Interface

April 2019
Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.
Contents

1 Introduction ........................................................................................................................................ 5
   Overview ........................................................................................................................................ 5
   Audience ....................................................................................................................................... 5
   Revision History ......................................................................................................................... 5

2 History ............................................................................................................................................. 7

3 General Layer Transmission Considerations .............................................................................. 11
   Data Bytes Format ....................................................................................................................... 11
   Data Types ................................................................................................................................... 11
   Monetary Characters ................................................................................................................... 11
   Codepages with IFCS’s XML ......................................................................................................... 12
   Overview ....................................................................................................................................... 12
   Framing .......................................................................................................................................... 13
   Client/Server ................................................................................................................................ 13
   Connection Mode ........................................................................................................................ 13
   Data Bytes Format ....................................................................................................................... 14
   Link Control Bytes Format ............................................................................................................ 14
   Basic Communication Description ............................................................................................... 14
   Full Duplex Link Control Bytes .................................................................................................... 14
   Other Notes ................................................................................................................................... 14
   WorkstationId ............................................................................................................................... 15
   Multi-Threading ............................................................................................................................ 15
   Schemas .......................................................................................................................................... 15

4 Fields and Records ....................................................................................................................... 20
   General Overview ........................................................................................................................ 20
   Communications and Link Control ................................................................................................. 20
      Link Description ....................................................................................................................... 20
      LinkAlive ................................................................................................................................ 21
      Sanity check ............................................................................................................................. 21
   Posting Records .......................................................................................................................... 22
      Inquiries .................................................................................................................................. 22
      PostInquiry ............................................................................................................................... 22
      PostList / PostListItem ............................................................................................................. 23
      PostRequest .............................................................................................................................. 25
      PostAnswer .............................................................................................................................. 28
      Notes to specific attributes ...................................................................................................... 28
5 Examples and Case Studies

<table>
<thead>
<tr>
<th>RequestType</th>
<th>AnswerStatus</th>
</tr>
</thead>
<tbody>
<tr>
<td>POSTINGS</td>
<td>Inquiries</td>
</tr>
<tr>
<td>Inquiries for all guests starting with &quot;M&quot;</td>
<td>Correct Posting</td>
</tr>
<tr>
<td>Wrong inquiry for all guests starting with &quot;M&quot;</td>
<td>Postrequest for 'Dareos', 4th match of the above list</td>
</tr>
<tr>
<td>Inquiry for room number &quot;123&quot;</td>
<td>Postrequest for 'Dareos', 4th match of the above list but MatchfromPostList is set incorrectly</td>
</tr>
<tr>
<td>Inquiry for room number &quot;217&quot;</td>
<td>Posting, where the sum of all Subtotals does not match</td>
</tr>
<tr>
<td>Inquiry for not checked in RoomNumber</td>
<td>This Posting, would be denied as follows:</td>
</tr>
<tr>
<td>Inquiry by Track2 of a magnetic key-card</td>
<td>Cumulating of subtotals</td>
</tr>
<tr>
<td>Inquiry to two Hotel</td>
<td>Posting of a Transaction where Inquiry is done using Track2 information (for example, magnetic Stripe card or UDID of a Key Card)</td>
</tr>
<tr>
<td>POSTINGS</td>
<td>Negative Posting (Void a check)</td>
</tr>
<tr>
<td>Negative Posting (Void a check)</td>
<td>Void Guest check from a room</td>
</tr>
<tr>
<td>Void Guest check from a room</td>
<td>PostRequest for 'Dareos', 4th match of the above list to void this transaction</td>
</tr>
<tr>
<td>PostRequest for 'Dareos', 4th match of the above list to void this transaction</td>
<td>Negative Cash/Credit card transaction</td>
</tr>
<tr>
<td>Negative Cash/Credit card transaction</td>
<td>Direct Posting to the Accounts Receivable Module</td>
</tr>
<tr>
<td>Direct Posting to the Accounts Receivable Module</td>
<td>Splitting of a check</td>
</tr>
</tbody>
</table>

Appendix A – Code Tables

<table>
<thead>
<tr>
<th>Code Tables</th>
<th>AnswerStatus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix A – Code Tables</td>
<td>RequestType</td>
</tr>
<tr>
<td>AnswerStatus</td>
<td>RequestType</td>
</tr>
</tbody>
</table>
1 Introduction

Overview

The purpose of this document is to set a standard for application record formats and data flows to be used for data communications between a Oracle Hospitality Property Management System (PMS) and a Hotel POS system (*). It gives a general description of record formats and data flow requirements and covers specifics for Record Types, Field Types, and Field usage.

For information regarding the low-level protocol specification and recommendations used by Oracle, please refer to the Oracle Interface Protocol Specification - Section of this document.

Note: XML-POS is supported by the following PMS systems from Oracle Hospitality:

Oracle Hospitality Suite 8 (any version) & Oracle Hospitality OPERA PMS (>= Ver. 5.x)

Audience

Point of Sale Vendors, integrating to Oracle PMS Interface using XML_POS Protocol.

Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>October 28, 2017</td>
<td>• Initial publication and conversion from former CHM file</td>
</tr>
</tbody>
</table>
| February 27, 2018| • Corrected typo in CreditLimitOverride attribute (Wrong: CreditLimitOverride*)  
                  • Removed ProfileId from PostRequest and PostAnswer record as it was never supported |
| May 08, 2018    | • Added Section **Timeouts waiting for responses in Fields and Records** |
March 2019

- Outlined HotelId format in PostList message
- Smaller adjustments in Multithreading section. All requests must go through one single TCP connection.
## 2 History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version number and change description</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 2007</td>
<td>Version 0.99 – first draft, overview, Record and Field Types</td>
</tr>
<tr>
<td>November 2007</td>
<td>Version 0.99b smaller corrections Added PaymentMethod to PostListItem</td>
</tr>
<tr>
<td>July 2008</td>
<td>Version 0.99d Removed &quot;LinkEnd&quot; documentation. This record has never been supported.</td>
</tr>
<tr>
<td>August 2008</td>
<td>Version 0.99e changed length of &quot;ResponseText&quot; in PostAnswer from 80 to 1000</td>
</tr>
<tr>
<td>November 2008</td>
<td>Version 0.99f smaller corrections and enhancements for better clarification</td>
</tr>
<tr>
<td>November 2008</td>
<td>Version 0.99g Corrected typo: SubTotal -&gt; Subtotal</td>
</tr>
<tr>
<td>January 2009</td>
<td>Version 0.99h clarification for attribute HotelId clarification about A/R postings clarification for length definition for RevenueCenter</td>
</tr>
<tr>
<td>February 2009</td>
<td>Version 0.99i Added description for STX/ETX characters</td>
</tr>
<tr>
<td>March 2009</td>
<td>Version 0.99j removed obsolete attributes from PostRequest examples</td>
</tr>
<tr>
<td>April 2009</td>
<td>Version 0.99l Changed allowed values for 'MaximumReturnedMatches' in Post inquiry to &gt;&quot;1&quot;</td>
</tr>
<tr>
<td>July 2009</td>
<td>Version 0.99m smaller corrections in descriptions</td>
</tr>
<tr>
<td>January 2010</td>
<td>Version 0.99n clarification for mandatory fields of PostAnswer in case AnswerStatus &lt;&gt;'OK'</td>
</tr>
<tr>
<td>January 2010</td>
<td>Version 1.00 - RELEASE</td>
</tr>
<tr>
<td>Date</td>
<td>Version number and change description</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>March 2010</td>
<td>Version 1.01 - corrected a typo in PostList table (Lastname-&gt;LastN)</td>
</tr>
<tr>
<td>July 2010</td>
<td>Version 1.02 - smaller corrections in global descriptions</td>
</tr>
<tr>
<td>January 2011</td>
<td>Version 1.03 - corrected length description for RomNumber attribute</td>
</tr>
<tr>
<td>January 2011</td>
<td>Version 1.04 - further description to posting records</td>
</tr>
<tr>
<td>June 2011</td>
<td>Version 1.05 - clarification for MatchfromPostList</td>
</tr>
<tr>
<td>July 2011</td>
<td>Version 1.06 - clarification split checks</td>
</tr>
<tr>
<td>September 2011</td>
<td>Version 1.07 - corrected typo in PostInquiry description Workstationid -&gt; WorkstationId</td>
</tr>
<tr>
<td>February 2012</td>
<td>Version 1.08 - Added LinkStart as sanity check</td>
</tr>
<tr>
<td>November 2012</td>
<td>Version 1.09 - Clarification for HotelId</td>
</tr>
<tr>
<td>April 2013</td>
<td>Version 1.10 - Added 'PostingDescription to PostRequest</td>
</tr>
<tr>
<td>January 2014</td>
<td>Version 2.00</td>
</tr>
<tr>
<td></td>
<td>Added fields</td>
</tr>
<tr>
<td></td>
<td>Reference3...Reference10</td>
</tr>
<tr>
<td></td>
<td>ServiceCharge5...ServiceCharge16</td>
</tr>
<tr>
<td>March 2014</td>
<td>Version 2.01 - Corrected length-description for HotelId</td>
</tr>
<tr>
<td>April 2014</td>
<td>Version 2.02</td>
</tr>
<tr>
<td></td>
<td>Changed length of FirstName to 80 characters</td>
</tr>
<tr>
<td></td>
<td>Changed length of LastName to 200 characters</td>
</tr>
<tr>
<td>May 2014</td>
<td>Version 2.03</td>
</tr>
<tr>
<td></td>
<td>Corrected description of &quot;LinkStart&quot;</td>
</tr>
<tr>
<td>July 2014</td>
<td>Version 2.04</td>
</tr>
<tr>
<td></td>
<td>Smaller corrections to field sizes</td>
</tr>
<tr>
<td>July 2014</td>
<td>Version 2.05</td>
</tr>
<tr>
<td></td>
<td>Corrected field size for Inquiry information</td>
</tr>
<tr>
<td>October 2015</td>
<td>Version 2.06</td>
</tr>
<tr>
<td></td>
<td>Corrected description Communications / Connection Mode</td>
</tr>
<tr>
<td>Date</td>
<td>Version number and change description</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>March 2016</td>
<td>Version 2.07</td>
</tr>
<tr>
<td></td>
<td>corrected &quot;0&quot; value for Tax items in PostRequest</td>
</tr>
<tr>
<td></td>
<td>Tax field ID description (was 0.00 which is not expected)</td>
</tr>
<tr>
<td></td>
<td>adjusted explanation of requirement sending all Subtotal, Discount, ServiceCharge,Tax attributes even no splitting is used</td>
</tr>
<tr>
<td></td>
<td>added examples for void/refund of transactions</td>
</tr>
<tr>
<td>August 2016</td>
<td>Version 2.08</td>
</tr>
<tr>
<td></td>
<td>Corrected CreditLimit Attribute Type in General Transmission Layer Considerations --&gt; Other Notes.</td>
</tr>
<tr>
<td>September 2016</td>
<td>Version 2.09</td>
</tr>
<tr>
<td></td>
<td>Corrected field length for PaymentMethod attribute in PostRequest record description</td>
</tr>
<tr>
<td>November 2016</td>
<td>Version 2.10</td>
</tr>
<tr>
<td></td>
<td>Added explanation of the PaymentMethod attribute values expected in the record types.</td>
</tr>
<tr>
<td>April 2017</td>
<td>Version 2.11</td>
</tr>
<tr>
<td></td>
<td>adjusted XML Schema format samples in 'Other Notes' section</td>
</tr>
<tr>
<td></td>
<td>adjusted PostRequest Field description: attributes Subtotal1-16, Tax1-16, Discount1-6, ServiceCharge1-16, RevenueCenter, ServingTime, Tip, Covers are no longer Required fields, but optional</td>
</tr>
<tr>
<td>May 2017</td>
<td>Version 2.12</td>
</tr>
<tr>
<td></td>
<td>adjusted Data Type section code description</td>
</tr>
<tr>
<td></td>
<td>adjusted Data Format columns in Record ID Types</td>
</tr>
<tr>
<td>October 2017</td>
<td>Version 2.13</td>
</tr>
<tr>
<td></td>
<td>corrected direction info for LinkAlive message at initialization and for Sanity Check</td>
</tr>
</tbody>
</table>
3 General Layer Transmission Considerations

Data Bytes Format

Records are composed of data bytes and link control bytes. The data portion of a record should not contain any bytes normally reserved for link control (Hex 00 through Hex 1F, and Hex 7F).

In order to avoid any confusion with code pages and the display of special characters, all characters≥ HEX 7F should be transferred as their XML-entity (for example, as &#xe4;).

For most transmissions, the rest of the standard ASCII character set is sufficient (Hex 20 through HEX 7E). Data is passed in an unpacked format; it should not be packed in ‘nibblized’, BCD, or other formats. This is to simplify installation and support.

Data Types

In general, fields are either numeric (decimal digits '0' - '9'), or alpha (all alphabetic letters). Some fields require some combination of these types.


ANS - Complete character set of used Code Page (all printable character), No Control Character.

N - Numeric characters, includes '0' - '9', the minus sign ('-') as leading character, and where necessary 'A' - 'F' and 'a' - 'f' as hex characters. These fields always reflect integer values (no decimal positions). No Alpha character, No Control character.

M - Monetary characters, includes all numeric characters and period (‘.’) as decimal indicators where necessary when coming FROM the PMS. Record send TO the PMS must be without a decimal indicator. The PMS can handle monetary fields without an implied decimal point depending on the regional/local setting and currency.

D - Date, numeric characters, formatted as YYMMDD

T - Time, numeric characters, formatted as HHMMSS

Note: As the PMS sends and expects to receive fields without padding, leading zeroes or spaces in all alphanumeric fields are considered significant data (that is, if a room number contains a leading zero or a leading space, this digit is regarded as part of the room number).

Monetary Characters

Posting details are generally expected without decimal indicators. For example, "1000" is translated to €10.00 while parsing an incoming posting record; granting the currency of the PMS is set to EURO and the decimals of the PMS are configured to 2-digits.

The definition of the used currency and the defined decimals is done within the PMS.
**Codepages with IFC8's XML**

Some programming environments have difficulties to properly process/transfer UTF-8 encoded Unicode-characters (multi-byte data) on physical Communication-Layers.

Meaning: transfer these characters over Socket without corruption.

Subsequently, IFC8 keeps data UTF-8 compatible, but it does not transfer multi-byte characters. However, the ability to receive multi-byte characters is supported.

Example:

*Turquoise* is the internal record, *Red* is the format as it is sent to the vendor system:

```
[11.03/11:57:38] => <GuestdataChange RoomNum="388" GuestNum="34769"
GuestName="Häusle" GuestFormattedname="Häusle" GuestLanguage="DE"
GuestShareflag="N" RequestType="0"/>

[11.03/11:57:38] < - <GuestdataChange RoomNum="388" GuestNum="34769"
GuestName="Häusle" GuestFormattedname="Häusle" GuestLanguage="DE"
GuestShareflag="N" RequestType="0"/>
```

The *GuestName="Häusle"* contains the Umlaut "ä" which is encoded as [E4] in UNI-CODE. In UTF-8 this is already coded as "Doublebyte" [C3,A4].

Using the XML-compatible Entity of Unicode-Characters in the format H&#x... [here "E4"&#xE4;] which is used for "ä"] will eliminate the problem of sending Multi-Byte characters without violating UTF-8 compatibility.

Taking this into account, one could say that no character > 127 (7F) would be transferred. This corresponds to the ASCII Code-page and thus the XML-Declaration `<xml version="1.0" encoding="ASCII""> would be theoretically valid.

Problems with this handling are NOT related to the used code-page and can mostly be found in the inability of an XML-Parser to resolve XML-compatible references as described above. (This problem has been observed with JAVA-classes.)

The transformation from UTF-8 to Unicode/Multibyte form is automatically handled during the loading of an XML-document in XML-DOM or SAX parsers (like MsXML*.DLL)

Examples:

```
<GuestdataChange RoomNum="388" GuestNum="34769" GuestName="Häusle"/>
```

Result when opened with IE:

```
<GuestdataChange RoomNum="388" GuestNum="34769" GuestName="Häusle" />
```

Communications

The communication is based on TCP/IP, and the exchange is based on easy-to-understand xml.

**Overview**

As TCP/IP is a robust protocol that supports its own handshaking, transmission integrity checking. The intention of this document is not to describe how TCP/IP works. However, there are some options that Oracle has chosen to implement in a specific manner. For example, Oracle uses stream type TCP/IP, not datagram. These implementation details are listed below.
Framing

Many applications that use TCP/IP simply send unframed data (that is, they don’t use link control bytes). However, since the low-level handshaking is performed by the TCP/IP stack, it is possible that a second data block may be sent before the receiving application has actually retrieved the first data block, even in a stream implementation. This results in a situation where the receiver has to be able to parse the data to insure that trailing data blocks are not accidentally discarded.

One approach is to implement a handshake at the application level; this can slow the throughput which defeats one of the main gains of using TCP/IP. The other method is to add start and end of data link control bytes as a mechanism to clearly mark a record. Even if the application receives more than a single data block, it can easily parse out the first complete block and process it, storing any extra characters until it can locate another complete block. Oracle therefore requires the use of link control bytes even with TCP/IP.

Client/Server

The interface program acts as the server for the TCP/IP connection. A port, or service, number is configured at installation; this is the port to which the client should try to connect. The interface program listens indefinitely on that port for a connection request. Upon receiving a request, the listening thread accepts it and starts a communication thread for the data exchange on a different port. The communication thread reads any data received and checks at least once every 10 milliseconds whether there is something to be sent.

When the connection is closed by the client, the interface keeps on listening for a new connection request.

Other systems that want to communicate using TCP/IP should implement a client that establishes one connection and uses this for all communications.

**Note:** Only one client at a time can have a connection for a given port number, and the Oracle interface communicates to a single application ONLY.

After a successful connection, the interface keeps listening on the given port. Should another connect request be received, the interface drops the original connection and connects to the client, which connected last.

This behavior is designed to allow a vendor to reestablish a broken link without restarting the Oracle Interface.

However, as restarting the link may take several seconds, it is recommended to keep an established connection open.

Connection Mode

The connection is built by the client and should be left up and running until one of the systems needs to close it for maintenance purposes. The client cannot connect, send out a transaction (either PostRequest / PostAnswer or PostInquiry/PostList/PostRequest/PostAnswer), and then close the connection again. As
stated above, it is expected that all communication is being done through one single connection.

**Data Bytes Format**

Records are composed of data bytes and link control bytes. The data portion of a record should not contain any bytes normally reserved for link control (Hex 00 through Hex 1B, and Hex 7F). The control characters from Hex 1C through Hex 1F are used by some systems as field separators; other control characters are sometimes used as data (such as CR - Hex 0D and HT - Hex 0A) to indicate display formatting. When this is the case, these characters are considered part of the normal data stream and do not require a preceding escape character; they are not available for use as link control characters.

For most transmissions, the rest of the standard ASCII character set is sufficient (Hex 20 through Hex 7E); however, in order to support multiple alphabets, extended ASCII (Hex 80 - Hex FF) may be used. Data is passed in an unpacked format; it should not be packed in nibblized, BCD or other formats. (Using non-standard ASCII characters require the corresponding XML-replacement. For example, `&` would be transferred as `&amp;`)

**Link Control Bytes Format**

Link control bytes are used to indicate the beginning and end of records and provide other functionality such as flow control. Control bytes fall into the range from Hex 00 through Hex 1B (though Hex 7F is also a control byte). Many of these have standard definitions, though some usages are determined by interface protocol specifications. The implementation and use of specific control bytes is described below.

**Basic Communication Description**

There is no line bid unless one system has sent an application level request to stop the link or has disconnected the TCP-socket (this is used only when the link is in an idle state). The link is always considered active.

The beginning of a record is marked by an `<STX>`. The data in the record is then sent in ASCII format, and the record terminates with an `<ETX>`.

**Full Duplex Link Control Bytes**

- `<STX>` (Hex 02) - `<STX>` is used to mark the beginning of a record. It is immediately followed by the data portion of the record.
- `<ETX>` (Hex 03) - `<ETX>` is used to mark the end of the data portion; however, it is not part of the data.

**Other Notes**

Low-level ACK/NAK responses are not required but can be supplied. Application level responses are always necessary.

In most cases where records are rejected at the application level, there is an application level response. For example, if a posting record is received correctly but contains bad/invalid data (that is, unknown room number), the application response would contain: `AnswerStatus="NG" ResponseText="INVALID ROOM."`
As the protocol is XML-based, all fields can be transferred in any order.

---

**NOTE:** Fields in response records from the PMS can be transferred in any order too. This can differ based on the PMS version and may change at any time.

---

XML does not have field length limitations. However, the connected database systems may have limitations for certain data. Subsequently, a format limitation has been defined for some fields.

Not all fields must be transferred in responses from the PMS. For example, FirstName would not be transferred if the first name of a guest is not available.

**WorkstationId**

The “WorkstationId” is the key attribute for all records. The PMS system stores the last inquiry of each and every workstation and compares a PostRequest with the last known inquiry.

Subsequently, it is NOT possible to send two Postings to only one inquiry.

**Multi-Threading**

Multiple requests from DIFFERENT workstations can be sent at any time. The responses can easily be found back through the “WorkstationId” and “SequenceNumber” attributes. There is no limit to the number of workstations.

All requests are channeled through a single TCP-connection.

**Schemas**

XML-Schemas for this protocol are available. However, they should ONLY be used to correctly import the messages. **The final record must always be sent WITHOUT link to a schema.**

---

**Note:** The following schemas are only samples in order to allow vendors to import. They might not contain current defined type format. Final reference for the schema attribute format can be found in "Fields and Records" --> "Posting Records" respective in Section "Data Types" and in related example sections.
<?xml version="1.0" encoding="UTF-8" ?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
    targetNamespace="PostInquirySchema_1" elementFormDefault="qualified">
  <element name="PostInquiry">
    <complexType>
      <attribute name="InquiryInformation" type="string" use="required" />
      <attribute name="MaximumReturnedMatches" type="string" use="optional" />
      <attribute name="SequenceNumber" type="string" use="required" />
      <attribute name="RequestType" type="string" use="required" />
      <attribute name="PaymentMethod" type="string" use="required" />
      <attribute name="Date" type="string" use="required" />
      <attribute name="Time" type="string" use="required" />
      <attribute name="RevenueCenter" type="string" use="optional" />
      <attribute name="WaiterId" type="string" use="optional" />
      <attribute name="WorkstationId" type="string" use="required" />
    </complexType>
  </element>
</schema>

<?xml version="1.0" encoding="UTF-8" ?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
    targetNamespace="PostListSchema_0" elementFormDefault="qualified">
  <element name="PostList">
    <complexType>
      <sequence>
        <element ref="PostListItem" maxOccurs="unbounded" />
      </sequence>
      <attribute name="HotelId" type="string" use="required" />
      <attribute name="PaymentMethod" type="string" use="required" />
      <attribute name="RevenueCenter" type="string" use="optional" />
      <attribute name="WorkstationId" type="string" use="required" />
      <attribute name="Date" type="string" use="required" />
      <attribute name="Time" type="string" use="required" />
    </complexType>
  </element>
</schema>

<?xml version="1.0" encoding="UTF-8" ?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
    targetNamespace="PostListItem_1" elementFormDefault="qualified">
  <element name="PostListItem">
    <complexType>
      <attribute name="RoomNumber" type="string" use="required" />
      <attribute name="ReservationId" type="string" use="required" />
      <attribute name="LastName" type="string" use="required" />
      <attribute name="Vip" type="string" use="optional" />
      <attribute name="PaymentMethod" type="string" use="optional" />
      <attribute name="NoPost" type="string" use="required" />
      <attribute name="CreditLimit" type="string" use="required" />
      <attribute name="ProfileId" type="string" use="required" />
      <attribute name="HotelId" type="string" use="required" />
      <attribute name="FirstName" type="string" use="optional" />
      <attribute name="Title" type="string" use="optional" />
      <attribute name="Reference1" type="string" use="optional" />
      <attribute name="Reference2" type="string" use="optional" />
      <attribute name="Reference3" type="string" use="optional" />
      <attribute name="Reference4" type="string" use="optional" />
      <attribute name="Reference5" type="string" use="optional" />
      <attribute name="Reference6" type="string" use="optional" />
    </complexType>
  </element>
</schema>
<complexType>
  <attribute name="Reference7" type="string" use="optional" />
  <attribute name="Reference8" type="string" use="optional" />
  <attribute name="Reference9" type="string" use="optional" />
  <attribute name="Reference10" type="string" use="optional" />
</complexType>
</element>
</schema>

<?xml version="1.0" encoding="UTF-8" ?>
<schema
xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="PostRequestSchema_I"
elementFormDefault="qualified">
  <element name="PostRequest">
    <complexType>
      <attribute name="RoomNumber" type="string" use="required" />
      <attribute name="ReservationId" type="string" use="required" />
      <attribute name="LastName" type="string" use="required" />
      <attribute name="HotelId" type="string" use="required" />
      <attribute name="RequestType" type="string" use="required" />
      <attribute name="InquiryInformation" type="string" use="required" />
      <attribute name="MatchFromPostList" type="string" use="required" />
      <attribute name="SequenceNumber" type="string" use="required" />
      <attribute name="TotalAmount" type="string" use="required" />
      <attribute name="CreditLimitOverride" type="string" use="optional" />
      <attribute name="PostingDescription" type="string" use="optional" />
      <attribute name="PaymentMethod" type="string" use="required" />
      <attribute name="Covers" type="string" use="optional" />
      <attribute name="RevenueCenter" type="string" use="optional" />
      <attribute name="ServingTime" type="string" use="optional" />
      <attribute name="CheckNumber" type="string" use="required" />
      <attribute name="Subtotal1" type="string" use="optional" />
      <attribute name="Subtotal2" type="string" use="optional" />
      <attribute name="Subtotal3" type="string" use="optional" />
      <attribute name="Subtotal4" type="string" use="optional" />
      <attribute name="Subtotal5" type="string" use="optional" />
      <attribute name="Subtotal6" type="string" use="optional" />
      <attribute name="Subtotal7" type="string" use="optional" />
      <attribute name="Subtotal8" type="string" use="optional" />
      <attribute name="Subtotal9" type="string" use="optional" />
      <attribute name="Subtotal10" type="string" use="optional" />
      <attribute name="Subtotal11" type="string" use="optional" />
      <attribute name="Subtotal12" type="string" use="optional" />
      <attribute name="Subtotal13" type="string" use="optional" />
      <attribute name="Subtotal14" type="string" use="optional" />
      <attribute name="Subtotal15" type="string" use="optional" />
      <attribute name="Subtotal16" type="string" use="optional" />
      <attribute name="Discount1" type="string" use="optional" />
      <attribute name="Discount2" type="string" use="optional" />
      <attribute name="Discount3" type="string" use="optional" />
      <attribute name="Discount4" type="string" use="optional" />
      <attribute name="Discount5" type="string" use="optional" />
      <attribute name="Discount6" type="string" use="optional" />
      <attribute name="Discount7" type="string" use="optional" />
      <attribute name="Discount8" type="string" use="optional" />
      <attribute name="Discount9" type="string" use="optional" />
      <attribute name="Discount10" type="string" use="optional" />
      <attribute name="Discount11" type="string" use="optional" />
      <attribute name="Discount12" type="string" use="optional" />
  </complexType>
</element>
</schema>
<complexType>
  <attribute name="AnswerStatus" type="string" use="optional" />
  <attribute name="RoomNumber" type="string" use="required" />
  <attribute name="ReservationId" type="string" use="required" />
  <attribute name="LastName" type="string" use="required" />
  <attribute name="AnswerStatus" type="string" use="required" />
  <attribute name="ResponseText" type="string" use="required" />
  <attribute name="CheckNumber" type="string" use="required" />
  <attribute name="SequenceNumber" type="string" use="required" />
  <attribute name="HotelId" type="string" use="required" />
  <attribute name="Date" type="string" use="required" />
</complexType>
</element>
</schema>
<attribute name="Time" type="string" use="required" />
<attribute name="PaymentMethod" type="string" use="required" />
<attribute name="RevenueCenter" type="string" use="required" />
<attribute name="WaiterId" type="string" use="optional" />
<attribute name="WorkstationId" type="string" use="required" />
</complexType>
</element>
</schema>
General Overview

The following pages contain descriptions of all supported record types and fields. The fields per record type are listed as table like:

<table>
<thead>
<tr>
<th>Record ID</th>
<th>Field ID</th>
<th>Description</th>
<th>Format</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostInquiry</td>
<td>Date</td>
<td></td>
<td>D</td>
<td>To Pms</td>
</tr>
<tr>
<td></td>
<td>InquiryInformation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MaximumReturnedMatches</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PaymentMethod</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RequestType</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RevenueCenter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SequenceNumber</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>WaiterId</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Workstationid</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field IDs marked in RED are mandatory and must be used at any time.
Field IDs marked in BLACK are optional and may be used. (If an optional field is not used, then the attribute as such must be omitted.)

Communications and Link Control

These Record Types are used to control the status of the link. The PMS only opens or closes the link when starting or stopping its software. The following records are used to define the current status of the link.

**Link Description**

<table>
<thead>
<tr>
<th>Record ID</th>
<th>Field ID</th>
<th>Description</th>
<th>Format</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinkDescription</td>
<td>Date</td>
<td>System Date</td>
<td>D</td>
<td>To Pms</td>
</tr>
<tr>
<td></td>
<td>Time</td>
<td>System Time</td>
<td>T</td>
<td>To Pms</td>
</tr>
<tr>
<td></td>
<td>VerNum</td>
<td>Software version of the POS system</td>
<td>ANS, max 20</td>
<td>To Pms</td>
</tr>
</tbody>
</table>

Example:

```xml
<LinkDescription Date="070818" Time="190649" VerNum="1.0" />
```
The LinkDescription record is used to signal to the PMS on the application layer level that the POS system is now ready to transmit data.

### LinkAlive

<table>
<thead>
<tr>
<th>Record ID</th>
<th>Field ID</th>
<th>Description</th>
<th>Format</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinkAlive</td>
<td>Date</td>
<td>System Date</td>
<td>D</td>
<td>From Pms</td>
</tr>
<tr>
<td></td>
<td>Time</td>
<td>System Time</td>
<td>T</td>
<td>From Pms</td>
</tr>
</tbody>
</table>

Example:

```
<<LinkAlive Date="070818" Time="071151"/>>
```

The LinkAlive record is used to signal to the POS on the application layer level that the PMS system is now ready to receive data. **No** data may be passed to the PMS before the LinkAlive record has been received. LinkAlive is passed in response to LinkDescription. However, as the LinkAlive signals that the PMS interface has initialized, this record can take several seconds to be transmitted after the LinkDescription has been received. Any record sent to the PMS prior to receiving the LinkAlive record is ignored.

### Sanity check

In order to check if the connection is still valid (that is, not closed by an active component like a firewall), the POS can send a LinkStart record:

(This functionality requires a minimum interface version of 8.7.4.6)

<table>
<thead>
<tr>
<th>Record ID</th>
<th>Field ID</th>
<th>Description</th>
<th>Format</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinkStart</td>
<td>Date</td>
<td>System Date</td>
<td>D</td>
<td>To Pms</td>
</tr>
<tr>
<td></td>
<td>Time</td>
<td>System Time</td>
<td>T</td>
<td>To Pms</td>
</tr>
</tbody>
</table>

Example:

```
<LinkStart Date="070818" Time="190649"/>
```
Posting Records

General
All posting transaction follow the same rule:
POS Inquiry
PMS Response to inquiry
POS posting with reference to the inquiry response
PMS posting response

Inquiries

Inquiries can be done for a room number, a guest’s name (full or in fraction) or by the contents of a magnetic stripe (Track2) of a guest’s key.
Inquiries only return a match on those guests who are currently checked in to the hotel.
Based on the inquiry info, multiple matches can be returned.

PostInquiry

<table>
<thead>
<tr>
<th>Record ID</th>
<th>Field ID</th>
<th>Description</th>
<th>Format</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostInquiry</td>
<td>Date</td>
<td>Date when the POS guest Check was opened</td>
<td>D</td>
<td>To PMS</td>
</tr>
<tr>
<td></td>
<td>InquiryInformation</td>
<td>Value to search for</td>
<td>ANS, max 20</td>
<td>To PMS</td>
</tr>
<tr>
<td></td>
<td>MaximumReturnedMatches</td>
<td>Maximum number of successful matches to be returned from the PMS</td>
<td>N, max. 2</td>
<td>To PMS</td>
</tr>
<tr>
<td></td>
<td>PaymentMethod</td>
<td>POS / Vendor payment method ID</td>
<td>ANS, max 6</td>
<td>To PMS</td>
</tr>
</tbody>
</table>
Positive Response to the Inquiry is the PostList record. (PostAnswer will be sent in case of a negative response.)

Maintain

**PostList / PostListItem**

<table>
<thead>
<tr>
<th>Record ID</th>
<th>Field ID</th>
<th>Description</th>
<th>Format</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostList</td>
<td>Date</td>
<td>System Date</td>
<td>D</td>
<td>From PMS</td>
</tr>
<tr>
<td>HotelId</td>
<td>ID(s) of the Hotel(s) where the response was generated</td>
<td></td>
<td>AN</td>
<td>From PMS</td>
</tr>
<tr>
<td>PaymentMethod</td>
<td>PaymentMethod received in PostInquiry from POS</td>
<td></td>
<td>ANS, max 6</td>
<td>From PMS</td>
</tr>
<tr>
<td>RevenueCenter</td>
<td>Revenue Center as received in PostInquiry from POS</td>
<td></td>
<td>N, max 5</td>
<td>From PMS</td>
</tr>
<tr>
<td>SequenceNumber</td>
<td>Sequence Number for this specific record, same as received in PostInquiry</td>
<td></td>
<td>N, max 6</td>
<td>From PMS</td>
</tr>
<tr>
<td>Field ID</td>
<td>Description</td>
<td>Format</td>
<td>Direction</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>System Time</td>
<td>T</td>
<td>From PMS</td>
<td></td>
</tr>
<tr>
<td>WaiterId</td>
<td>POS Waiter definition, same as received in PostInquiry</td>
<td>AN, max 20</td>
<td>From PMS</td>
<td></td>
</tr>
<tr>
<td>WorkstationId</td>
<td>POS workstation ID, same as received in PostInquiry (normally: Computername)</td>
<td>ANS, max 15</td>
<td>From PMS</td>
<td></td>
</tr>
<tr>
<td>PostList()</td>
<td>Matches, may occur multiple times</td>
<td>n/a</td>
<td>From PMS</td>
<td></td>
</tr>
</tbody>
</table>

A PostList record contain at least one PostList|Item-tag - and may contain many more, depending on the amount of matches found, based on the inquiry.

The **HotelID** in the **PostList** header will contain all Hotel-IDs of the PostList|tem, separated by 'pipe' (HEX 7C). Each contained HotelId is listed once.

<table>
<thead>
<tr>
<th>Record ID</th>
<th>Field ID</th>
<th>Description</th>
<th>Format</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostListItem</td>
<td>CreditLimit</td>
<td>Current available credit balance on the guest’s folio, can be &quot;0&quot; or a negative value if the credit-limit has already been reached</td>
<td>M - max 15</td>
<td>From PMS</td>
</tr>
<tr>
<td>FirstName</td>
<td></td>
<td>Guest’s first name</td>
<td>ANS, max 80</td>
<td>From PMS</td>
</tr>
<tr>
<td>HotelId</td>
<td></td>
<td>ID of the Hotel where the guest was found</td>
<td>N, max 1</td>
<td>From PMS</td>
</tr>
<tr>
<td>LastName</td>
<td></td>
<td>Guest’s last name</td>
<td>ANS, max 200</td>
<td>From PMS</td>
</tr>
<tr>
<td>NoPost</td>
<td></td>
<td>This flag defines if posting to the guest’s account is allowed or not.</td>
<td>'0' = Posting allowed '1' = Posting not allowed</td>
<td>From PMS</td>
</tr>
</tbody>
</table>
### PaymentMethod

**PaymentMethod**

PMS Reservation Payment method (not POS Payment Method)

ANS, max 40 From PMS

### ProfileId

**ProfileId**

ID of the guest’s profile

N, max 10 From PMS

### Reference1 .... Reference10

10 configurable reference fields, used to transfer information about a guest

ANS, max 30 From PMS

### ReservationId

**ReservationId**

ID of the guest’s reservation

N, max 10 From PMS

### RoomNumber

**RoomNumber**

Room number of the guest

ANS, max 15 From PMS

### Title

**Title**

Guest’s title

ANS, max 10 From PMS

### Vip

**Vip**

Guest VIP status, normally a numeric value, where ‘0’= non-VIP

ANS, max 10 From PMS

---

### PostRequest

<table>
<thead>
<tr>
<th>Record ID</th>
<th>Field ID</th>
<th>Description</th>
<th>Format</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostRequest</td>
<td>CheckNumber</td>
<td>POS Guest Check number</td>
<td>N, max 10</td>
<td>To PMS</td>
</tr>
<tr>
<td></td>
<td>Covers</td>
<td>Number of people on this guest check</td>
<td>N, max 3</td>
<td>To PMS</td>
</tr>
<tr>
<td></td>
<td>CreditLimitOverride</td>
<td>Flag which defines, if the PMS Credit limit should be ignored for this posting. Normally only allowed for POS supervisors.</td>
<td>&quot; &lt;blank&gt; = don’t override 'N' = don’t override 'Y' = override</td>
<td>To PMS</td>
</tr>
<tr>
<td></td>
<td>Date</td>
<td>Date when the POS check was opened</td>
<td>D</td>
<td>To PMS</td>
</tr>
<tr>
<td>Record ID</td>
<td>Field ID</td>
<td>Description</td>
<td>Format</td>
<td>Direction</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>-------------</td>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>Discount1 ... Discount16</td>
<td>Up to 16 different discounts can be passed to the PMS. NOTE: value must be negative!!</td>
<td>N, max 15</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td>HotelId</td>
<td>HotelId is found in PostListItem</td>
<td>N, max 1</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td>InquiryInformation</td>
<td>Inquiry Information as passed in the corresponding inquiry (Must match, posting will be refused if not.)</td>
<td>ANS, max 15</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td>LastName</td>
<td>Guest’s name as found in PostListItem</td>
<td>ANS, max 200</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td>MatchfromPostList 1</td>
<td>Number of the PostListItem (see examples)</td>
<td>N, max 3</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td>PaymentMethod</td>
<td>POS / Vendor payment method ID</td>
<td>ANS, max 6</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td>PostingDescription</td>
<td>Text-description of this posting. Can be used in the PMS for the folio-printout, so it cannot be longer than 20 characters</td>
<td>ANS, max 20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RequestType</td>
<td>see RequestType Table</td>
<td>N, max 4</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td>ReservationId</td>
<td>Guest’s Reservation ID as found in PostListItem</td>
<td>N, max 10</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td>RevenueCenter 2</td>
<td>POS Revenue Center</td>
<td>N, max 5</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td>RoomNumber</td>
<td>Guest’s Room Number</td>
<td>ANS, max 15</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td>Record ID</td>
<td>Field ID</td>
<td>Description</td>
<td>Format</td>
<td>Direction</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>-------------</td>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td><strong>SequenceNumber</strong></td>
<td>Sequence number for this record, will be echoed back in PMSs response</td>
<td>N, max 6</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td><strong>ServiceCharge1 ... ServiceCharge16</strong></td>
<td>Up to 16 Service Charges</td>
<td>N, max 15</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td><strong>ServingTime 3</strong></td>
<td>Serving Time (For example: '1' = breakfast, '2' = lunch, '3' = afternoon tea, '4' = dinner)</td>
<td>N, max 2</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td><strong>Subtotal1 ... Subtotal16</strong></td>
<td>Up to 16 Subtotals</td>
<td>N, max 15</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td><strong>Tax1 ... Tax16</strong></td>
<td>Up to 16 Taxes (Only used in countries with Add-On tax)</td>
<td>N, max 15</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>Time when the POS guest Check was opened</td>
<td>T</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td><strong>Tip</strong></td>
<td>Tip amount</td>
<td>N, max 15</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td><strong>TotalAmount</strong></td>
<td>Total of the check, must be the exact sum of all Subtotals, Discounts, ServiceCharges, Tax &amp; Tip</td>
<td>N, max 15</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td><strong>WaiterId</strong></td>
<td>POS Waiter ID</td>
<td>ANS, max 20</td>
<td>To PMS</td>
<td></td>
</tr>
<tr>
<td><strong>WorkstationId</strong></td>
<td>POS workstation ID (normally: Computername)</td>
<td>ANS, max 15</td>
<td>To PMS</td>
<td></td>
</tr>
</tbody>
</table>

1. Mandatory for Room postings, for Cash & Credit card postings set to MatchfromPostList=""
2. Mandatory for postings where the subtotals will be sent as per Revenue Center split
3. Mandatory for postings where the subtotals will be sent as per Serving Time split
# PostAnswer

<table>
<thead>
<tr>
<th>Record ID</th>
<th>Field ID</th>
<th>Description</th>
<th>Format</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostAnswer</td>
<td>AnswerStatus</td>
<td>Response code to the related request see AnswerStatus table</td>
<td>ANS, max 2</td>
<td>From PMS</td>
</tr>
<tr>
<td>CheckNumber 1, 2</td>
<td>POS Guest Check number</td>
<td></td>
<td>N, max 10</td>
<td>From PMS</td>
</tr>
<tr>
<td>Date</td>
<td>System Date</td>
<td></td>
<td>D</td>
<td>From PMS</td>
</tr>
<tr>
<td>HotelId</td>
<td>HotelId where charge was posted to</td>
<td></td>
<td>N, max 1</td>
<td>From PMS</td>
</tr>
<tr>
<td>LastName 1,2</td>
<td>Guest's last Name</td>
<td></td>
<td>ANS, max 200</td>
<td>From PMS</td>
</tr>
<tr>
<td>PaymentMethod</td>
<td>POS / Vendor payment method ID</td>
<td></td>
<td>ANS, max 6</td>
<td>From PMS</td>
</tr>
<tr>
<td>ReservationId 1, 2</td>
<td>Guest's Reservation ID</td>
<td></td>
<td>N, max 10</td>
<td>From PMS</td>
</tr>
<tr>
<td>ResponseText</td>
<td>Reason why a posting has been posted or why it was denied (in dependency to AnswerStatus)</td>
<td></td>
<td>ANS, max 1000</td>
<td>From PMS</td>
</tr>
<tr>
<td>RevenueCenter</td>
<td>POS Revenue Center</td>
<td></td>
<td>N, max 5</td>
<td>From PMS</td>
</tr>
<tr>
<td>RoomNumber 1, 2</td>
<td>Guest's room number</td>
<td></td>
<td>ANS, max 15</td>
<td>From PMS</td>
</tr>
<tr>
<td>SequenceNumber</td>
<td>Sequence number for this record, same as received in PostInquiry/PostRequest</td>
<td></td>
<td>N, max 6</td>
<td>From PMS</td>
</tr>
<tr>
<td>Time</td>
<td>System Time</td>
<td></td>
<td>T</td>
<td>From PMS</td>
</tr>
<tr>
<td>WaiterId</td>
<td>POS Waiter ID</td>
<td></td>
<td>ANS, max 20</td>
<td>From PMS</td>
</tr>
<tr>
<td>WorkstationId</td>
<td>POS workstation ID, same as received in PostInquiry/PostRequest (normally: Computername)</td>
<td></td>
<td>ANS, max 15</td>
<td>From PMS</td>
</tr>
</tbody>
</table>

- 1 only in responses to PostRequest
- 2 not sent in negative responses to PostInquiry

## Notes to specific attributes

**PaymentMethod:**

The **PaymentMethod attribute** is used in all record types, but it has two different meanings:

- **PaymentMethod in PostInquiry, PostRequest, PostList and in PostAnswer** record type shall be the Paymethod ID of the POS vendor system identifying the POS Payment Type (Room, Cash, Credit Card etc)
- **PaymentMethod** in PostListItem record type will contain the Payment Type ID of the Guest in the PMS System (Cash, Debit, Amex, VISA ect).

It is not expected that the PMS Payment Type value is sent in the PostRequest PaymentMethod attribute.

Examples:

```xml
[<PostInquiry InquiryInformation="101" MaximumReturnedMatches="16" SequenceNumber="1234" RequestType="4" PaymentMethod="16" Date="070905" Time="194121" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" />]
```

The above inquiry is for room number '101' from POS-workstation 'POS1.'

MaximumReturnedMatches identifies that no more than 16 matches should be returned from the PMS. This value is optional and should only be set by those POS systems which have difficulties to display many matches.

```xml
[<PostList SequenceNumber="1234" HotelId="1" PaymentMethod="16" RevenueCenters="1" WaiterId="Waiter_1" WorkstationId="POS1" Date="070906" Time="083017" />

<PostListItem RoomNumber="101" ReservationId="12345678" LastName="Koehler" FirstName="Andreas" Title="Mr." Vip="0" NoPost="0" CreditLimit="100.00" PaymentMethod="Diners Club" ProfileId="12345678" Reference1="Definable Value1" Reference2="Definable Value2" HotelId="1" />

</PostList>

[<PostRequest RoomNumber="101" ReservationId="12345678" ProfileId="12345678" LastName="Koehler" HotelId="1" RequestType="4" InquiryInformation="101" MatchFromPostList="1" SequenceNumber="1235" TotalAmount="20000" CreditLimitOverride="" PaymentMethod="16" Covers="1" RevenueCenter="1" ServingTime="1" CheckNumber="12345678" Subtotal1="1000" Subtotal2="1000" Subtotal3="1000" Subtotal4="1000" Tip="200" ServiceCharge1="100" ServiceCharge2="100" ServiceCharge3="100" ServiceCharge4="100" Tax1="160" Tax2="160" Tax3="160" Tax4="160" Date="070906" Time="082942" WaiterId="Waiter_1" WorkstationId="POS1" />

</PostRequest>]

[<PostAnswer RoomNumber="101" ReservationId="12345678" ProfileId="12345678" LastName="Koehler" AnswerStatus="OK" ResponseText="ClearText_Value" CheckNumber="12345678" SequenceNumber="1235" HotelId="1" Date="070906" Time="083031" PaymentMethod="16" RevenueCenter="1" WaiterId="Waiter_1" WorkstationId="POS1" />

</PostAnswer>]
```

### Revenue Fields:

There are multiple fields in the PostRequest record, where revenues can be transferred:

- SubTotal1 ... SubTotal16
- Discount1 ... Discount 16
- ServiceCharge1 ... ServiceCharge 16
- Tax1 ... Tax16
- Tip
- TotalAmount

**NOTE:** The global rule is as follows:

```
SubTotals+Discounts+ServiceCharges+Tip+Taxes = TotalAmount
```

All postings are denied when the above does not match.
**SubTotals:**
These fields are used to transfer the main group totals of the respective guest check.
A typical setup is as follows:
SubTotal1 = Food
SubTotal2 = Alcoholic Beverages
SubTotal3 = Nonalcoholic Beverages
SubTotal4 = Miscellaneous revenues
Up to SubTotal16
(These values are configurable in the PMS upon installation time.)

**Discounts:**
The discounts are in direct relation to the SubTotals and should always contain negative values.
Typically this would mean the following:
Discount1 = Food Discount
Discount2 = Alcoholic Beverages Discount
Discount3 = Nonalcoholic Beverages Discount
Discount4 = Miscellaneous revenues Discount
Up to Discount16
Alternatively, the discounts can be deducted from the respective SubTotal value. In that case, the Discount field should be set to “0.”

**ServiceCharges:**
ServiceCharges should be filled with revenue (for example, Room Service extra charges).

**Taxes:**
Taxes should only be populated in countries with Add-on tax. In countries with inclusive VAT, always Tax-fields do not need to be sent or filled with “0.”

**Tip:**
This field should be filled with Tips. These revenues are treated as paid-out/non-revenue transactions (=normally non-taxable, depending on the local laws).

**Date & Time:**
It is recommended to fill Date & Time with the time-stamp when a guest-check was opened.

---

**Note:** If multiple postings are sent for the same check, Date & Time and Check Number **MUST** have the same value in all postings.

Timeouts waiting for responses:

The PMS responses to a PostInquiry or PostRequest message may vary and might take longer due to the complexity of the request or with bigger sites with high data traffic. The external system should implement proper time outs while waiting for the response messages for the above commands.

If an external system does not receive the expected response message by PMS it should time out at this stage and stop waiting for a response.

The minimum default time out should be 30 seconds for PostInquiry and PostRequest commands. It is recommended to make the time out value configurable to fit every single case.
5 Examples and Case Studies

The examples in this chapter are listed without STX/ETX.

Inquiries

Inquiry for all guests starting with "M"

(RequestType is set to '8', as the POS only wants matches from names.)

```xml
<PostInquiry InquiryInformation="M" MaximumReturnedMatches="16"
SequenceNumber="1235" RequestType="8" PaymentMethod="16" Date="070905" Time="194121" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" />
```

Response with 16 matches. More than 16 matches could not be transferred as the POS has limited the maximum results to 16 by setting `MaximumReturnedMatches="16"`.

```xml
<PostList SequenceNumber="1235" HotelId="1" PaymentMethod="16" RevenueCenter="1"
WaiterId="Waiter1" WorkstationId="POS1" Date="070905" Time="194121" />

<PostListItem RoomNumber="106" ReservationId="33750" LastName="Mros"
FirstName="AndrŠ #x26;E9;" Title="Mr." NoPost="0" CreditLimit="-56.35"
ProfileId="6552" HotelId="1" />

<PostListItem RoomNumber="111" ReservationId="33341" LastName="Martius"
FirstName="Christoph und Carlotta" Title="Mr. &amp; Mrs." NoPost="0"
CreditLimit="-1271" ProfileId="38974" HotelId="1" />

<PostListItem RoomNumber="117" ReservationId="33945" LastName="Müller"
FirstName="Karin" Title="Mrs." NoPost="0" CreditLimit="10" ProfileId="37856"
HotelId="1" />

<PostListItem RoomNumber="122" ReservationId="35739" LastName="Mishra"
NoPost="0" CreditLimit="322" ProfileId="39759" HotelId="1" />

<PostListItem RoomNumber="134" ReservationId="35031" LastName="Mair"
FirstName="Gabriele" Title="Mrs." Vip="001" NoPost="0" CreditLimit="200"
ProfileId="39243" HotelId="1" />

<PostListItem RoomNumber="231" ReservationId="34836" LastName="Ma"
FirstName="Baotian" Title="Mr." Vip="001" NoPost="0" CreditLimit="-180.6"
ProfileId="12674" HotelId="1" />

<PostListItem RoomNumber="306" ReservationId="31353" LastName="Muench"
FirstName="Bernold" Title="Mr." NoPost="0" CreditLimit="-974" ProfileId="37655"
HotelId="1" />

<PostListItem RoomNumber="350" ReservationId="33900" LastName="Marr"
FirstName="Sabine" Title="Mrs." NoPost="0" CreditLimit="-504" ProfileId="8023"
HotelId="1" />

<PostListItem RoomNumber="370" ReservationId="34470" LastName="Maubach"
FirstName="Lisa" Title="Mrs." NoPost="0" CreditLimit="100" ProfileId="38795"
HotelId="1" />

<PostListItem RoomNumber="387" ReservationId="23116" LastName="Meyer"
FirstName="Holger" Title="Mr." NoPost="0" CreditLimit="-671.8" ProfileId="9372"
HotelId="1" />

<PostListItem RoomNumber="398" ReservationId="35030" LastName="Maleschka"
FirstName="Birgit" Title="Mrs." NoPost="0" CreditLimit="296" ProfileId="39242"
HotelId="1" />

<PostListItem RoomNumber="399" ReservationId="27241" LastName="Muench"
FirstName="Bernold" Title="Mr." NoPost="0" CreditLimit="100" ProfileId="37655"
HotelId="1" />

<PostListItem RoomNumber="417" ReservationId="33464" LastName="Müller"
FirstName="Karin" Title="Mrs." NoPost="0" CreditLimit="87.8" ProfileId="37856"
HotelId="1" />
```
Wrong inquiry for all guests starting with "M"

The following inquiry is wrong because the POS wants to get all guests whose last names start with "M" but sets the RequestType to "4," which signals a search for a Room number:

```xml
<PostInquiry InquiryInformation="M" MaximumReturnedMatches="16" SequenceNumber="1239" RequestType="4" PaymentMethod="16" Date="070905" Time="194121" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" />
```

```xml
<PostAnswer AnswerStatus="NG" ResponseText="Room not found" SequenceNumber="1239" HotelId="1" Date="070905" Time="194121" PaymentMethod="16" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" />
```

**Note:** Unsuccessful inquiries are responded to with PostAnswer.

Inquiry for room number "123"

```xml
<PostInquiry InquiryInformation="123" MaximumReturnedMatches="16" SequenceNumber="1236" RequestType="4" PaymentMethod="16" Date="070905" Time="194121" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" />
```

Single match as return:

```xml
<PostList SequenceNumber="1236" HotelId="1" PaymentMethod="16" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" Date="070905" Time="194121"> 
</PostList>
```

Inquiry for room number "217"

```xml
<PostInquiry InquiryInformation="217" MaximumReturnedMatches="16" SequenceNumber="1236" RequestType="4" PaymentMethod="16" Date="070905" Time="194121" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" />
```

Several matches are returned, as the room is occupied by more than one guest (called 'Sharing Reservations'):

```xml
<PostList SequenceNumber="1236" HotelId="1" PaymentMethod="16" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" Date="070905" Time="194121">
```

**NOTE:** Characters over decimal 126 are ALWAYS sent as their respective entity. (LastName="Müller" would be sent as: LastName="M&uuml;ller")
Inquiry for not checked in RoomNumber

Negative response:

Positive response:

Note: The Response Text can be modified at any time. NEVER trigger for a specific response text, but ALWAYS look at AnswerStatus to find the value which identifies if the response is positive or negative.

Inquiry by Track2 of a magnetic key-card

Positive response:

Note: Track2-data is ONLY used to identify a guest. A subsequent posting would be done using the data as supplied in the PostList record.

Inquiry to two Hotel

Note: PhoneNumbers are used only for PostList records.
The HotelID in the PostList header will contain all Hotel-IDs of the PostList item, separated by 'pipe' (HEX 7C). Each contained HotelID is listed once.

POSTINGS

Correct Posting

Inquiry

```xml
<PostInquiry InquiryInformation="D" MaximumReturnedMatches="16" SequenceNumber="1280" RequestType="14" PaymentMethod="16" DateofDate="070905" Time="194121" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" />

<PostList SequenceNumber="1280" HotelId="1" PaymentMethod="16" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" DateofDate="070905" Time="194121" />

<PostListItem RoomNumber="238" ReservationId="35746" LastName="De Sascha" NoPost="0" CreditLimit="50" ProfileId="39767" HotelId="1" />

<PostListItem RoomNumber="341" ReservationId="35797" LastName="Dido" FirstName="Kay" NoPost="0" CreditLimit="50" ProfileId="39785" HotelId="1" />

<PostListItem RoomNumber="365" ReservationId="34865" LastName="Doering" FirstName="Hagen" Title="Mr." Vip="001" NoPost="0" CreditLimit="-275.35" ProfileId="39106" HotelId="1" />

<PostListItem RoomNumber="464" ReservationId="31045" LastName="Dareos" FirstName="Jutta" Title="Mrs." NoPost="0" CreditLimit="-163" ProfileId="34886" HotelId="1" />

<PostListItem RoomNumber="9020" ReservationId="33772" LastName="Dalep" FirstName="Kryeziu" Title="Mr." NoPost="0" CreditLimit="1000237.85" ProfileId="21046" HotelId="1" />

<PostListItem RoomNumber="9006" ReservationId="33948" LastName="Dorff" FirstName="Joe" Title="Mr." NoPost="0" CreditLimit="999897" ProfileId="9872" HotelId="1" />

</PostList>
```

The result list has 6 matches. The posting must have a direct reference to one of the matches. The first PostListItem (De Sascha) can be referenced as MatchfromPostList="1". The last one (Dorff) from the list would be MatchfromPostList="6".

An incorrect link leads to a denied posting.

Postrequest for 'Dareos', 4th match of the above list

---

**Note:** Tax attributes are set to "0," as all other amounts are tax-inclusive. Only populate the tax-attributes in countries with add-on tax.

```xml
<PostRequest RoomNumber="464" ReservationId="31045" ProfileId="34886" LastName="Dareos" HotelId="1" RequestType="4" InquiryInformation="D" MatchfromPostList="1" SequenceNumber="1281" TotalAmount="12750" CreditLimitOverride="Y" PaymentMethod="16" Covers="4" RevenueCenter="1" ServingTime="1" CheckNumber="54586562" Subtotal1="8000" Subtotal2="3500" Discount="-1500" Tip="550" ServiceCharge1="2200" ServiceCharge2="0" Date="070906" Time="224700" WaiterId="Waiter_6" WorkstationId="POS1" />
```

Positive response:
<PostAnswer RoomNumber="464" ReservationId="31045" LastName="Dareos" AnswerStatus="OK" ResponseText="Posting successful" CheckNumber="54586562" SequenceNumber="1281" HotelId="1" Date="070906" Time="224700" PaymentMethod="16" RevenueCenter="1" WaiterId="Waiter_6" WorkstationId="POS1" />

Note: The ResponseText can be modified at any time. NEVER trigger for a specific response text but ALWAYS look at AnswerStatus to find the value which identifies if the response is positive or negative.

Postrequest for 'Dareos', 4th match of the above list but MatchfromPostList is set incorrectly

<PostRequest RoomNumber="464" ReservationId="31045" ProfileId="34886" LastName="Dareos" HotelId="1" RequestType="4" InquiryInformation="D" MatchfromPostList="5" SequenceNumber="1280" TotalAmount="12750" CreditLimitOverride="Y" PaymentMethod="16" Covers="4" RevenueCenter="1" ServingTime="1" CheckNumber="54586562" Subtotal1="8000" Subtotal2="3500" Discount1="1500" Discount2="0" Tip="550" ServiceCharge1="2200" Date="070906" Time="224700" WaiterId="Waiter_6" WorkstationId="POS1" />

Negative response:

<PostAnswer RoomNumber="464" ReservationId="31045" LastName="Dareos" AnswerStatus="UR" ResponseText="Unprocessable request" CheckNumber="54586562" SequenceNumber="1280" HotelId="1" Date="070906" Time="224700" PaymentMethod="16" RevenueCenter="1" WaiterId="Waiter_6" WorkstationId="POS1" />

Posting, where the sum of all Subtotals does not match

<PostRequest RoomNumber="498" ReservationId="33265" ProfileId="37241" LastName="Adameck" HotelId="1" RequestType="14" InquiryInformation="A" MatchfromPostList="4" SequenceNumber="1237" TotalAmount="4600" CreditLimitOverride="" PaymentMethod="16" Covers="1" RevenueCenter="1" ServingTime="1" CheckNumber="12345678" Subtotal1="1000" Subtotal2="1000" Subtotal3="1000" Subtotal4="1000" Tip="200" ServiceCharge1="100" ServiceCharge2="100" ServiceCharge3="100" ServiceCharge4="100" Tax1="160" Tax2="160" Tax3="160" Tax4="160" Date="070906" Time="082942" WaiterId="Waiter_1" WorkstationId="POS1" />

This Posting, would be denied as follows:

<PostAnswer RoomNumber="498" ReservationId="33265" LastName="Adameck" AnswerStatus="DM" ResponseText="Difference mismatch" CheckNumber="12345678" SequenceNumber="1237" HotelId="1" Date="070906" Time="082942" PaymentMethod="16" RevenueCenter="1" WaiterId="Waiter_1" WorkstationId="POS1" />

Cumulating of subtotals

The reason for this denial is the sums in all subtotal attributes do not match the total amount. (In this case: Taxes have not been calculated into the TotalAmount.)

Total Amount = 46.00
SubTotal1 = 10.00
| SubTotal2   | 10.00 |
| SubTotal3   | 10.00 |
| SubTotal4   | 10.00 |
| Tip         | 2.00  |
| ServiceCharge1 | 1.00   |
| ServiceCharge2 | 1.00   |
| ServiceCharge3 | 1.00   |
| ServiceCharge4 | 1.00   |
| Tax1        | 1.60  |
| Tax2        | 1.60  |
| Tax3        | 1.60  |
| Tax4        | 1.60  |

Sum of Subtotals, Tip, Service Charges and Taxes = 52.40

**Posting of a Transaction where Inquiry is done using Track2 information (for example, magnetic Stripe card or UDID of a Key Card)**

Ensure that the **InquiryInformation** value is matching the **InquiryInformation** value from the used **PostInquiry** request. Otherwise, **PostRequest** might be refused.

```xml
<PostInquiry InquiryInformation="1040022102035774" MaximumReturnedMatches="16" SequenceNumber="1236" RequestType="2" PaymentMethod="16" Date="070905" Times="194121" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" />
```

Positive response:

```xml
<PostList SequenceNumber="1250" HotelId="1" PaymentMethod="16" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" Date="070905" Times="194121" />
<PostListItem RoomNumber="223" ReservationId="34834" LastName="Chan" FirstName="Kenneth Chung Leung" Title="Mr." NoPost="0" CreditLimit="-198.2" ProfileId="39077" HotelId="1" />
</PostList>
```

```xml
<PostRequest RoomNumber="223" ReservationId="34834" ProfileId="39077" LastName="Chan" HotelId="1" RequestType="4" InquiryInformation="1040022102035774" MatchfromPostList="1" SequenceNumber="1237" TotalAmount="4600" CreditLimitOverride="" PaymentMethod="16" Covers="1" RevenueCenter="1" ServingTimes="1" CheckNumber="12345678" Subtotal1="1000" Subtotal2="1000" Subtotal3="1000" Subtotal4="1000" Discount1="" Discount2="" Discount3="" Discount4="" Tip="200" ServiceCharge1="100" ServiceCharge2="100" ServiceCharge3="100" ServiceCharge4="100" Taxi="160" Tax2="160" Tax3="160" Tax4="160" Date="070905" Times="194122" WaiterId="Waiter_1" WorkstationId="POS1" />
```

```xml
<PostAnswer RoomNumber="223" ReservationId="34834" LastName="Chan" AnswerStatus="OK" ResponseText="Request successful" CheckNumber="12345678" SequenceNumber="1238" HotelId="1" Date="070905" Times="194123" PaymentMethod="16" RevenueCenter="1" WaiterId="Waiter_6" WorkstationId="POS1" />
```

**Posting of a cash/credit card transaction**

The difference between a cash posting and a room posting is identified through the **PaymentMethod**. At installation time, all Room-payment methods which are configured on the POS.

```xml
<PostRequest RoomNumber="" ReservationId="" ProfileId="" LastName="" HotelId="1" RequestType="1" InquiryInformation="" MatchfromPostList="" SequenceNumber="1290" TotalAmount="12750" CreditLimitOverride="N" PaymentMethod="35" Covers="4" RevenueCenter="1" ServingTimes="1" CheckNumber="54586562" Subtotal1="8000" Subtotal2="3500" Discount1="-1500" Tip="550" ServiceCharge1="2200" Date="070906" Times="224700" WaiterId="Waiter_6" WorkstationId="POS1" />
```
Note: All guest related attributes can stay empty for transfer of cash/credit card revenues, yet the attributes are mandatory and need to be set. RequestType for cash/credit card transaction must be set to “1.”

HotelId: The default value should be “1.” The value may be different if one POS system serves several hotels through just one interface. Please contact your local Oracle Hospitality dealer for details of this enhanced handling.

<PostAnswer RoomNumbers="9400" ReservationId="28633" LastName="POS Cash Account" AnswerStatus="OK" ResponseText="Request successful" CheckNumber="54586562" SequenceNumber="1290" HotelId="1" Date="070905" Time="224700" PaymentMethod="35" RevenueCenters="1" WaiterId="Waiter_6" WorkstationId="POS1" />

Negative Posting (Void a check)

Void Guest check from a room

If a transaction has to be voided / refunded from a Room because the wrong payment method or the wrong items have been charged to the room, the whole guest check-in must be voided from the PMS.
Handling is the same as for regular Room transaction:

Send the Room Inquiry first (while awaiting the PostList/PostListItem) to verify the guest is still checked in.

Inquiry

<PostInquiry InquiryInformation="D" MaximumReturnedMatches="16" SequenceNumber="1285" RequestType="14" PaymentMethod="16" Date="070905" Time="194121" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" />

PostList

<PostList SequenceNumber="1285" HotelId="1" PaymentMethod="16" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" Date="070905" Time="194121" />
  <PostListItem RoomNumber="238" ReservationId="35746" LastName="De Sascha" NoPost="0" CreditLimit="50" ProfileId="39767" HotelId="1" />
  <PostListItem RoomNumber="341" ReservationId="35797" LastName="Dido" FirstName="Kay" NoPost="0" CreditLimit="50" ProfileId="39785" HotelId="1" />
  <PostListItem RoomNumber="365" ReservationId="34865" LastName="Doering" FirstName="Hagen" Title="Mr." Vip="001" NoPost="0" CreditLimit="-275.35" ProfileId="39106" HotelId="1" />
  <PostListItem RoomNumber="464" ReservationId="31045" LastName="Dareos" FirstName="Jutta" Title="Mrs." NoPost="0" CreditLimit="-163" ProfileId="34866" HotelId="1" />
  <PostListItem RoomNumber="9020" ReservationId="33772" LastName="Dalip" FirstName="Kryeziu" Title="Mr." NoPost="0" CreditLimit="1000237.85" ProfileId="21046" HotelId="1" />
  <PostListItem RoomNumber="9006" ReservationId="33948" LastName="Dorff" FirstName="Joe" Title="Mr." NoPost="0" CreditLimit="999897" ProfileId="9872" HotelId="1" />
</PostList>
Next, send PostRequest to send all items but send the related amounts with a negative sign.

**PostRequest for 'Dareos', 4th match of the above list to void this transaction**

```xml
<PostRequest RoomNumber="464" ReservationId="31045" ProfileId="34886"
LastName="Dareos" HotelId="1" RequestType="4" InquiryInformation="D"
MatchfromPostList="4" SequenceNumber="1286" TotalAmount="-12750"
CreditLimitOverride="Y" PaymentMethod="16" Covers="-4" RevenueCenter="1"
ServingTime="1" CheckNumber="54586562" Subtotal1="-8000" Subtotal2="-3500"
Discount="1500" Tip="-550" ServiceCharge="-2200" Date="070906" Time="194121"
WaiterId="Waiter1" WorkstationId="POS1" />
```

TotalAmount, Subtotals, ServiceCharges, and Tip amounts must contain a negative sign ("-").
In this case, Discount amounts must not have a negative ("-"), sign.
Covers must be sent with a negative sign ("-").

**Negative Cash/Credit card transaction**

To void a guest check for a Cash / Credit card payment, just send the related PostRequest Record with exact same values as for the positive transaction, but also using related negative amounts and negative Covers as for the room posting.

```xml
<PostRequest RoomNumber="" ReservationId="" ProfileId="" LastName="" HotelId="1"
RequestType="1" InquiryInformation="" MatchfromPostList="" SequenceNumber="1291"
TotalAmount="-12750" CreditLimitOverride="N" PaymentMethod="35" Covers="-4"
RevenueCenter="1" ServingTime="1" CheckNumber="54586562" Subtotal1="-8000"
Subtotal2="-3500" Discount="1500" Tip="-550" ServiceCharge="-2200" Date="070906"
Time="224700" WaiterId="Waiter_6" WorkstationId="POS1" />
```

TotalAmount, Subtotals, ServiceCharges, and Tip amounts must contain a negative sign ("-").
In this case, Discount amounts must not have a negative ("-"), sign.
Covers must be sent with a negative sign ("-").

**Direct Posting to the Accounts Receivable Module**

Some Oracle Hospitality PMS systems allow direct postings to its A/R module.
In order to do so, a second Room Payment method needs to be configured on both the POS-system and the PMS system. Inquiries can be done by name or by account-number.

**Splitting of a check**

If the other system is a POS that supports splitting checks between guests or payment methods, the individual subtotals, taxes, and discounts should also be split so that when added together, they equal the Total Amount to be posted. This way, all rounding corrections are handled by the same system, and the revenue totals between the POS and the PMS match.
Appendix A – Code Tables

AnswerStatus

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO</td>
<td>Posting denied because overwriting the CreditLimit is not allowed</td>
</tr>
<tr>
<td>DM</td>
<td>Sum of Subtotals does not match TotalAmount</td>
</tr>
<tr>
<td>FX</td>
<td>Posting denied because the guest is not allowed to use posting features</td>
</tr>
<tr>
<td>IA</td>
<td>Invalid Account information</td>
</tr>
<tr>
<td>NA</td>
<td>Posting Denied - Night-Audit procedure is currently running</td>
</tr>
<tr>
<td>NG</td>
<td>Guest/Room not found</td>
</tr>
<tr>
<td>NP</td>
<td>Posting denied because NoPost is set</td>
</tr>
<tr>
<td>OK</td>
<td>Posted successfully</td>
</tr>
<tr>
<td>UR</td>
<td>Unprocessable request (generic error)</td>
</tr>
</tbody>
</table>

"OK" is the only positive response to a posting. All other statuses signal that the posting was unsuccessful. In the case of AnswerStatus "OK," the POS should display the contents of ResponseText to the POS user. This text normally explains the reason for the denied posting. The text is in English by default and can be translated to other languages depending on the PMS version.

RequestType

The RequestType is used to describe the inquiry information. It can signal that the POS system is currently looking for a guest-name, a room number, or that it passed the contents of a magnetic stripe. The same logic applies to usage in the PostRequest function.

The values are bit-oriented and can be combined if the POS cannot decide.

The logic behind this functionality is to speed up the searches in the PMS databases.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Posting of Cash/Creditcard (PostRequest only)</td>
</tr>
<tr>
<td>2</td>
<td>Inquiry by Track2 of a magnetic stripe</td>
</tr>
<tr>
<td>4</td>
<td>Inquiry by RoomNumber</td>
</tr>
<tr>
<td>8</td>
<td>Inquiry by guest's name</td>
</tr>
</tbody>
</table>

- The following values for inquiries are valid:
  2, 4, 8 if a clear distinction can be made

- Or combinations:
  6 = Inquiry by Track2 or room number, contents of InquiryInformation cannot be a last name
10 = Inquiry by Track2 or guest’s name, contents of InquiryInformation cannot be a room number
12 = Inquiry by room number or guest’s name, contents of InquiryInformation cannot be the contents of a magnetic stripe
14 = Inquiry by Track2 or guest’s name or room number

If the POS cannot make a decision, it should always send the inquiry with RequestType="14" if magnetic stripes are supported. Otherwise, it should send RequestType="12."

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please include your name, address, email address, and telephone number (optional). If you need assistance with Oracle software, please contact your support representative or Oracle Support Services.

If you require training or instruction in Oracle software, please contact your local Oracle office and inquire about Oracle University offerings. A list of Oracle offices is available on our web site at www.oracle.com.