

Oracle® Hospitality  
Hotel Property  
Interface  
IFC8 XML\_POS  
Specification



Release 2.18

October 2020

ORACLE®

Copyright ©, 2020, 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 about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

# Contents

Preface	5
<hr/>	
1 History	7
<hr/>	
2 General Layer Transmission Considerations	11
<hr/>	
Data Bytes Format	11
Data Types	11
Monetary Characters	12
Codepages with IFC8's XML	12
Physical Transmission Layer:	13
TCP/IP	13
Overview	13
TCP/IP SSL Stream	13
Framing	14
Client/Server	14
Connection Mode	15
Data Bytes Format	15
Link Control Bytes Format	15
Basic Communication Description	16
Full Duplex Link Control Bytes	16
Other Notes	16
WorkstationId	16
Multi-Threading	17
Schemas	17
<hr/>	
3 Fields and Records	21
<hr/>	
General Overview	21
Communications and Link Control	21
Link Description	21
LinkAlive	22
Sanity check	22
Posting Records	23
Inquiries	23
PostInquiry	23
PostList / PostListItem	24
PostRequest	26
PostAnswer	28

Notes to specific attributes	28
SubTotals:	30
Discounts:	31
ServiceCharges:	31
Taxes:	31
Tip:	32
Date & Time:	32
Timeouts waiting for responses:	32
<b>4 Examples and Case Studies</b>	<b>33</b>
<hr/>	
Inquiries	33
Inquiry for all guests starting with "M"	33
Wrong inquiry for all guests starting with "M"	34
Inquiry for room number "123"	34
Inquiry for room number "217"	34
Inquiry for not checked in RoomNumber	35
Inquiry by Track2 of a magnetic key-card	35
Inquiry to two Hotel	36
POSTINGS	36
Correct Posting	36
Postrequest for 'DName4', 4th match of the above list	37
Postrequest for 'DName4', 4th match of the above list but MatchfromPostList is set incorrectly	37
PostingRequest containing POS Cash Rounding Difference amount	38
Posting, where the sum of all Subtotals does not match	38
Cumulating of subtotals	39
Posting of a Transaction where Inquiry is done using Track2 information (for example, magnetic Stripe card or UDID of a Key Card)	39
Posting of a cash/credit card transaction	40
Negative Posting (Void a check)	40
Void Guest check from a room	40
Negative Cash/Credit card transaction	42
Direct Posting to the Accounts Receivable Module	42
Splitting of a check	42
<b>5 Appendix A – Code Tables</b>	<b>43</b>
<hr/>	
AnswerStatus	43
RequestType	43

# Preface

## **Purpose**

This guide explains 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.

## **Audience**

This document is intended for Point of Sale Vendors, integrating to Oracle PMS Interface using XML\_POS Protocol.

## **Customer Support**

To contact Oracle Customer Support, access My Oracle Support at the following URL:

<https://support.oracle.com>

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

## **Documentation**

Oracle Hospitality product documentation is available on the Oracle Help Center at

<http://docs.oracle.com/en/industries/hospitality/>

**Table 1 Revision History**

Date	Description
October 28, 2017	<ul style="list-style-type: none"><li>Initial publication and conversion from former CHM file</li></ul>
February 27, 2018	<ul style="list-style-type: none"><li>Corrected typo in <b>CreditLimitOverride</b> attribute (Wrong: <b>CreditLimitOverride*</b>)</li><li>Removed <b>ProfileId</b> from PostRequest and PostAnswer record as it was never supported</li></ul>
May 08, 2018	<ul style="list-style-type: none"><li>Added Section <b>Timeouts waiting for responses</b> in <b>Fields and Records</b></li></ul>
March 2019	<ul style="list-style-type: none"><li>Outlined <b>HotelId</b> format in <b>PostList</b> message</li><li>Smaller adjustments in Multithreading section. All requests must go through one single TCP connection.</li></ul>
May 2020	<ul style="list-style-type: none"><li>Removed incorrect ProfileId attributes from sample messages PostRequest and PostAnswer</li><li>Physical Transmission Layer: Added TCP/IP SSL stream connection using certificates</li><li>Added <b>Subtotal17</b> attribute in PostRequest message. Used to send POS Cash rounding difference amount (i.e. when occurring in Postings with rounded Total Amount containing discounts)</li><li>Added more details on handling of Posting Sequence Number in Posting messages</li></ul>
October 2020	<ul style="list-style-type: none"><li>Ver.2.18 Adjusted the Connection mode pointing out that the communication between vendor system and PMS Interface is designed to be contained within a local area network infrastructure</li></ul>

# 1 History

Date	Version number and change description
September 2007	Version 0.99 – first draft, overview, Record and Field Types
November 2007	Version 0.99b smaller corrections Added PaymentMethod to PostListItem
July 2008	Version 0.99d Removed "LinkEnd" documentation. This record has never been supported.
August 2008	Version 0.99e changed length of "ResponseText" in PostAnswer from 80 to 1000
November 2008	Version 0.99f smaller corrections and enhancements for better clarification
November 2008	Version 0.99g Corrected typo: SubTotal -> Subtotal
January 2009	Version 0.99h clarification for attribute HotelId clarification about A/R postings clarification for length definition for <i>RevenueCenter</i>
February 2009	Version 0.99i Added description for STX/ETX characters
March 2009	Version 0.99j removed obsolete attributes from PostRequest examples
April 2009	Version 0.99l Changed allowed values for 'MaximumReturnedMatches' in Post inquiry to >"1"
July 2009	Version 0.99m smaller corrections in descriptions
January 2010	Version 0.99n clarification for mandatory fields of PostAnswer in case AnswerStatus <>'OK'
January 2010	Version 1.00 - RELEASE
March 2010	Version 1.01 - corrected a typo in PostList table (Lastname->LastName)
July 2010	Version 1.02 - smaller corrections in global descriptions
January 2011	Version 1.03 - corrected length description for RomNumber attribute

<b>Date</b>	<b>Version number and change description</b>
January 2011	Version 1.04 - further description to posting records
June 2011	Version 1.05 - clarification for MatchfromPostList
July 2011	Version 1.06 - clarification split checks
September 2011	Version 1.07 - corrected typo in PostInquiry-description Workstationid -> WorkstationId
February 2012	Version 1.08 - Added LinkStart as sanity check
November 2012	Version 1.09 - Clarification for HotelId
April 2013	Version 1.10 - Added 'PostingDescription to PostRequest
January 2014	Version 2.00 Added fields Reference3...Reference10 ServiceCharge5...ServiceCharge16
March 2014	Version 2.01 - Corrected length-description for HotelId
April 2014	Version 2.02 Changed length of FirstName to 80 characters Changed length of LastName to 200 characters
May 2014	Version 2.03 Corrected description of "LinkStart"
July 2014	Version 2.04 Smaller corrections to field sizes
July 2014	Version 2.05 Corrected field size for Inquiry information
October 2015	Version 2.06 Corrected description Communications / Connection Mode
March 2016	Version 2.07 corrected "0" value for Tax items in PostRequest Tax field ID description (was 0.00 which is not expected) adjusted explanation of requirement sending all Subtotal, Discount, ServiceCharge,Tax attributes even no splitting is used added examples for void/refund of transactions
August 2016	Version 2.08 Corrected CreditLimit Attribute Type in General Transmission Layer Considerations --> Other Notes.
September 2016	Version 2.09 Corrected field length for PaymentMethod attribute in PostRequest record description



Date	Version number and change description
November 2016	Version 2.10 Added explanation of the PaymentMethod attribute values expected in the record types.
April 2017	Version 2.11 adjusted XML Schema format samples in 'Other Notes' section adjusted PostRequest Field description: attributes Subtotal1-16, Tax1-16, Discount1-6, ServiceCharge1-16, RevenueCenter, ServingTime, Tip, Covers are no longer Required fields, but optional
May 2017	Version 2.12 adjusted Data Type section code description adjusted Data Format columns in Record ID Types
October 2017	Version 2.13 corrected direction info for LinkAlive message at initialization and for Sanity Check



# 2 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  $\geq$  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.

**AN** - Alphanumeric characters. Correspond to the character set of ASCII Code Page with codes 32-127. No Control Character.

**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="Näme" GuestFormattedname="Näme" GuestLanguage="DE" GuestShareflag="N"
RequestType="0"/>
[11.03/11:57:38] <- <GuestdataChange RoomNum="388" GuestNum="34769"
GuestName="N&#xE4;me" GuestFormattedname="N&#xE4;me" GuestLanguage="DE"
GuestShareflag="N" RequestType="0"/>
```

The `GuestName="Näme"` 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"ä"&#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="N&#xE4;me"/>
```

Result when opened with IE:

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

```
<?xml version="1.0" encoding="UTF-8"?><GuestdataChange RoomNum="388"  
GuestNum="34769" GuestName="N&#xE4;me" />
```

Result when opened with IE:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<GuestdataChange RoomNum="388" GuestNum="34769" GuestName="Name" />  
Communications
```

## Physical Transmission Layer:

### TCP/IP

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.

### TCP/IP SSL Stream

In addition to above TCP/IP connection it is possible to establish a secured TCP/IP connection between the Interface application (**as of Version 8.14.5.0**) and the vendor system using Server/Client certificates.

At time of TCP/IP connection a server certificate (optional additional client certificate) will be validated between both parties.

Mutual authentication is required as a security level which ensures both parties will authenticate each other.

The certificates should be defined as \*.pfx and \*.crt files.

Certificates can be generated by Certification Authority (CA) or can be self-signed certificates.

As they will work as point-to-point connection validation only valid for these two parties, it is not necessary to import them to the Microsoft Cert store, even it is supported by the Interface application.

TCP/IP SSL Stream uses the same handling (Framing, Client/Server, Connection Mode) than the above described TCP/IP connection.

A sample TCP SSL Client project can be provided which will explain the alternative validation of self-signed certificates in file format via "RemoteCertificateValidationCallback" function without import in Microsoft's cert store. Contact the Oracle validation team for more details.

## 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 always 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 communication is designed to be contained within a local area network infrastructure and not for communication through active components over the internet.

A persistent connection to the ORACLE interface is always required. 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_I" 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_O" elementFormDefault="qualified">
  <element name="PostList">
    <complexType>
      <sequence>
        <element ref="PostListItem" maxOccurs="unbounded" />
      </sequence>
      <attribute name="SequenceNumber" type="string" use="required" />
      <attribute name="HotelId" type="string" use="required" />
      <attribute name="PaymentMethod" 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" />
      <attribute name="Date" type="string" use="required" />
      <attribute name="Time" type="string" use="required" />
    </complexType>
  </element>
  <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" />
      <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" />
    </complexType>
  </element>
</schema>

```

```

<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" />
<attribute name="Discount13" type="string" use="optional" />
<attribute name="Discount14" type="string" use="optional" />
<attribute name="Discount15" type="string" use="optional" />
<attribute name="Discount16" type="string" use="optional" />
<attribute name="Tip" type="string" use="optional" />
<attribute name="ServiceCharge1" type="string" use="optional" />
<attribute name="ServiceCharge2" type="string" use="optional" />
<attribute name="ServiceCharge3" type="string" use="optional" />
<attribute name="ServiceCharge4" type="string" use="optional" />
<attribute name="ServiceCharge5" type="string" use="optional" />
<attribute name="ServiceCharge6" type="string" use="optional" />
<attribute name="ServiceCharge7" type="string" use="optional" />
<attribute name="ServiceCharge8" type="string" use="optional" />
<attribute name="ServiceCharge9" type="string" use="optional" />
<attribute name="ServiceCharge10" type="string" use="optional" />
<attribute name="ServiceCharge11" type="string" use="optional" />
<attribute name="ServiceCharge12" type="string" use="optional" />
<attribute name="ServiceCharge13" type="string" use="optional" />
<attribute name="ServiceCharge14" type="string" use="optional" />
<attribute name="ServiceCharge15" type="string" use="optional" />
<attribute name="ServiceCharge16" type="string" use="optional" />
<attribute name="Tax1" type="string" use="optional" />

```

```

<attribute name="Tax2" type="string" use="optional" />
<attribute name="Tax3" type="string" use="optional" />
<attribute name="Tax4" type="string" use="optional" />
<attribute name="Tax5" type="string" use="optional" />
<attribute name="Tax6" type="string" use="optional" />
<attribute name="Tax7" type="string" use="optional" />
<attribute name="Tax8" type="string" use="optional" />
<attribute name="Tax9" type="string" use="optional" />
<attribute name="Tax10" type="string" use="optional" />
<attribute name="Tax11" type="string" use="optional" />
<attribute name="Tax12" type="string" use="optional" />
<attribute name="Tax13" type="string" use="optional" />
<attribute name="Tax14" type="string" use="optional" />
<attribute name="Tax15" type="string" use="optional" />
<attribute name="Tax16" type="string" use="optional" />
<attribute name="Date" type="string" use="required" />
<attribute name="Time" type="string" use="required" />
<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="PostAnswerSchema_0" elementFormDefault="qualified">
  <element name="PostAnswer">
    <complexType>
      <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" />
      <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>

```

# 3 Fields and Records

## General Overview

The following pages contain descriptions of all supported record types and fields.

The fields per record type are listed as table like:

Record ID	Field ID	Description	Format	Direction
PostInquiry	Date			
	InquiryInformation			
	MaximumReturnedMatches			
	PaymentMethod			
	RequestType			
	RevenueCenter			
	SequenceNumber			
	Time			
	WaiterId			
	Workstationid			

Field IDs marked in **RED** are mandatory and must be used at any time containing values. 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

Record ID	Field ID	Description	Format	Direction
LinkDescription	Date	System Date	D	To Pms
	Time	System Time	T	To Pms
	VerNum	Software version of the POS system	ANS, max 20	To Pms

Example:

```
<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

Record ID	Field ID	Description	Format	Direction
LinkAlive	Date	System Date	D	From Pms
	Time	System Time	T	From Pms

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)**

Record ID	Field ID	Description	Format	Direction
LinkStart	Date	System Date	D	To Pms
	Time	System Time	T	To Pms

Record ID	Field ID	Description	Format	Direction
LinkAlive	Date	System Date	D	From Pms
	Time	System Time	T	From Pms

Example:

<LinkStart Date="070818" Time="190649" />

The interface will respond with:

<LinkAlive Date="070818" Time="190650"/>

## Posting Records

### General

All posting transaction follow the same rule:

1. POS Inquiry
2. PMS Response to inquiry
3. POS posting with reference to the inquiry response
4. 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

Record ID	Field ID	Description	Format	Direction
PostInquiry	Date	Date when the POS guest Check was opened	D	To PMS
	InquiryInformation	Value to search for	ANS, max 20	To PMS
	MaximumReturnedMatches	Maximum number of successful matches to be returned from the PMS If used, the value must be >'1'. Systems using this protocol must be capable of selecting guests from multiple reservations.	N, max. 2	To PMS
	PaymentMethod	POS / Vendor payment method ID	ANS, max 6	To PMS

<b>RequestType</b>	Description of Inquiry Information see <a href="#">RequestType table</a>	N	To PMS
<b>RevenueCenter</b>	POS Revenue Center	N	To PMS
<b>SequenceNumber</b> <sup>1</sup>	Sequence Number for this specific record	N	To PMS
<b>Time</b>	Time when the POS guest Check was opened	T	To PMS
<b>WaiterId</b>	POS Waiter definition	AN, max 20	To PMS
<b>WorkstationId</b>	POS workstation ID (normally: Computername)	AN	To PMS

<sup>1</sup> Must be unique as per **PostInquiry** message sent.

Positive Response to the Inquiry is the **PostList** record.  
Negative response to the Inquiry is the **PostAnswer** record.

## PostList / PostListItem

Record ID	Field ID	Description	Format	Direction
PostList	<b>Date</b>	System Date	D	From PMS
	<b>HotelId</b>	ID(s) of the Hotel(s) where the response was generated	AN	From PMS
	<b>PaymentMethod</b>	PMS PaymentMethod	ANS, max 6	From PMS
	<b>RevenueCenter</b>	Revenue Center as received in PostInquiry from POS	N	From PMS
	<b>SequenceNumber</b>	Sequence Number for this specific record, same as received in PostInquiry	N	From PMS
	<b>Time</b>	System Time	T	From PMS
	<b>WaiterId</b>	POS Waiter definition, same as received in PostInquiry	AN, max 20	From PMS
	<b>WorkstationId</b>	POS workstation ID, same as received in PostInquiry (normally: Computername)	ANS	From PMS



<b>PostListItem()</b>	Matches, may occur multiple times	From PMS
-----------------------	-----------------------------------	----------

A PostList record contain at least one PostListItem 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 PostListitem, separated by 'pipe' (HEX 7C). Each contained HotelId is listed once.

Record ID	Field ID	Description	Format	Direction
PostListItem	<b>CreditLimit</b>	Current available credit balance on the guest's folio, can be "0" or a negative value if the credit-limit has already been reached	M - max 15	From PMS
	FirstName	Guest's first name	ANS, max 80	From PMS
	<b>HotelId</b>	ID of the Hotel where the guest was found	N, max 1	From PMS
	<b>LastName</b>	Guest's last name	ANS, max 200	From PMS
	<b>NoPost</b>	This flag defines if posting to the guest's account is allowed or not.	'0' = Posting allowed '1' = Posting not allowed	From PMS
	PaymentMethod	PMS Reservation Payment method (not POS Payment Method)	ANS, max 40	From PMS
	<b>ProfileId</b>	ID of the guest's profile	N,	From PMS
	Reference1 .... Reference10	10 configurable reference fields, used to transfer information about a guest	ANS, max 30	From PMS
	<b>ReservationId</b>	ID of the guest's reservation	N,	From PMS
	<b>RoomNumber</b>	Room number of the guest	ANS, max 15	From PMS
	Title	Guest's title	ANS, max 10	From PMS
	Vip	Guest VIP status, normally a numeric value, where '0'= non-VIP	ANS, max 10	From PMS

# PostRequest

Record ID	Field ID	Description	Format	Direction
PostRequest	<b>CheckNumber</b>	POS Guest Check number	N, max 10	To PMS
	Covers	Number of people on this guest check	N, max 3	To PMS
	CreditlimitOverride	Flag which defines, if the PMS Credit limit should be ignored for this posting. Normally only allowed for POS supervisors.	" <blank> = don't override 'N' = don't override 'Y' = override	To PMS
	<b>Date</b>	Date when the POS check was opened	D	To PMS
	Discount1 ... Discount16	Up to 16 different discounts can be passed to the PMS. NOTE: value must be negative!!	N, max 15	To PMS
	<b>HotellId</b>	HotellId is found in PostListItem	N, max 1	To PMS
	<b>InquiryInformation</b>	Inquiry Information as passed in the corresponding inquiry (Must match, posting will be refused if not.)	ANS, max 15	To PMS
	<b>LastName</b>	Guest's name as found in PostListItem	ANS, max 200	To PMS
	<b>MatchfromPostList</b> 1	Number of the PostListItem (see examples)	N, max 3	To PMS
	<b>PaymentMethod</b>	POS / Vendor payment method ID	ANS, max 6	To PMS
	PostingDescription	Text-description of this posting. Can be used in the PMS for the folio-printout, so it cannot be longer than 20 characters	ANS, max 20	
	<b>RequestType</b>	see <a href="#">RequestType Table</a>	N, max 4	To PMS
	<b>ReservationId</b>	Guest's Reservation ID as found in PostListItem	N,	To PMS

Record ID	Field ID	Description	Format	Direction
	RevenueCenter <b>2</b>	POS Revenue Center	N	To PMS
	<b>RoomNumber</b>	Guest's Room Number	ANS, max 15	To PMS
	<b>SequenceNumber</b> <b>5</b>	Sequence number for this record, will be echoed back in PMSs response	N,	To PMS
	ServiceCharge1 ... ServiceCharge16	Up to 16 Service Charges	N, max 15	To PMS
	ServingTime <b>3</b>	Serving Time (For example: '1' = breakfast '2' = lunch '3' = afternoon tea '4' = dinner)	N, max 2	To PMS
	Subtotal1 ... Subtotal16	Up to 16 Subtotals	N, max 15	To PMS
	Subtotal17 <b>4</b>	POS Cash Rounding Difference Amount	N,max 15	To PMS
	Tax1 ... Tax16	Up to 16 Taxes (Only used in countries with Add-On tax)	N, max 15	To PMS
	<b>Time</b>	Time when the POS guest Check was opened	T	To PMS
	Tip	Tip amount	N, max 15	To PMS
	TotalAmount	Total of the check, must be the exact sum of all Subtotals, Discounts, ServiceCharges, Tax & Tip	N, max 15	To PMS
	WaiterId	POS Waiter ID	ANS, max 20	To PMS
	WorkstationId	POS workstation ID (normally: Computername)	ANS, max 15	To PMS

- 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
- 4** **POS Cash Rounding Difference amount** only in case Subtotal1-16 already used. POS Cash rounding difference amount (i.e. when occurring in Cash Postings with rounded Total Amount containing discounts) – **Requires specific Oracle Hotel Property Interface version!**
- 5** Must be unique as per **PostRequest** message sent shall not be the value from previous corresponding **Posting Inquiry** message

# PostAnswer

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

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

## Notes to specific attributes

### SequenceNumber:

The Sequence number for a PostInquiry or PostRequest message. Generated by the vendor system.

Will be sent back in corresponding PostList or PostAnswer message

SequenceNumber shall be different in each Message. Not intended to be sent in PostInquiry and in corresponding PostRequest message.

Not intended to be generated as per POS workstation or Revenue Center, but as a global Number.

Can be reset with POS System's new business day.

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

Positive response:

```
<PostList SequenceNumber="1236" HotelId="1" PaymentMethod="16" RevenueCenter="1"
WaiterId="Waiter1" WorkstationId="POS1" Date="070905" Time="194121">
  <PostListItem RoomNumber="223" ReservationId="34834" LastName="Name"
  FirstName="First Name" Title="Mr." NoPost="0" CreditLimit="-198.2"
  ProfileId="39077" HotelId="1" />
</PostList>
```

Posting request:

```
<PostRequest RoomNumber="223" ReservationId="34834" LastName="Name" HotelId="1"
RequestType="4" InquiryInformation="1040022102035774" MatchfromPostList="1"
SequenceNumber="1237" TotalAmount="4600" CreditlimitOverride="" PaymentMethod="16"
Covers="1" RevenueCenter="1" ServingTime="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" Tax1="160"
Tax2="160" Tax3="160" Tax4="160" Date="070905" Time="194122" WaiterId="Waiter_1"
WorkstationId="POS1" />
  <PostAnswer RoomNumber="223" ReservationId="34834" LastName="Name"
  AnswerStatus="OK" ResponseText="Request successful" CheckNumber="12345678"
  SequenceNumber="1237" HotelId="1" Date="070905" Time="194123" PaymentMethod="16"
  RevenueCenter="1" WaiterId="Waiter_6" WorkstationId="POS1" />
```

### 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 Payment ID of the POS vendor system identifying the POS Payment Type (Room, Cash, Credit Card ect)

- **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:

```
<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.

```

    <PostList SequenceNumber="1234" HotelId="1" PaymentMethod="16"
RevenueCenter="1" WaiterId="Waiter_1" WorkstationId="POS1" Date="070906"
Time="083017">
    <PostListItem RoomNumber="101" ReservationId="12345678" LastName="Name"
FirstName="First Name" 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" LastName="Name"
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" />

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

```

### 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

**Subtotal17 = POS Cash Rounding Difference amount** (i.e. when occurring in Postings with rounded Total Amount containing discounts) when Subtotal1-16 already used for transfer of main article groups

**Requires specific Oracle Hotel Property Interface version!**

**POS Cash Rounding Difference amount** can also be send in Subtotal1-16 when not all of the 16 Subtotal fields are used.

(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.



# 4 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.)

```
<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"`.

```
<PostList SequenceNumber="1235" HotelId="1" PaymentMethod="16" RevenueCenter="1"
WaiterId="Waiter1" WorkstationId="POS1" Date="070905" Time="194121">
  <PostListItem RoomNumber="106" ReservationId="33750" LastName="M-Name1"
FirstName="FirstName" Title="Mr." NoPost="0" CreditLimit="-56.35" ProfileId="6552"
HotelId="1" />
  <PostListItem RoomNumber="111" ReservationId="33341" LastName="M-Name2"
FirstName="FirstName" Title="Mr. & Mrs." NoPost="0" CreditLimit="-1271"
ProfileId="38974" HotelId="1" />
  <PostListItem RoomNumber="117" ReservationId="33945" LastName="M-Name3"
FirstName="FirstName" Title="Mrs." NoPost="0" CreditLimit="10" ProfileId="37856"
HotelId="1" />
  <PostListItem RoomNumber="122" ReservationId="35739" LastName="M-Name4"
NoPost="0" CreditLimit="322" ProfileId="39759" HotelId="1" />
  <PostListItem RoomNumber="134" ReservationId="35031" LastName="M-Name5"
FirstName="FirstName" Title="Mrs." Vip="001" NoPost="0" CreditLimit="200"
ProfileId="39243" HotelId="1" />
  <PostListItem RoomNumber="231" ReservationId="34836" LastName="M-Name6"
FirstName="FirstName" Title="Mr." Vip="001" NoPost="0" CreditLimit="-180.6"
ProfileId="12674" HotelId="1" />
  <PostListItem RoomNumber="306" ReservationId="31353" LastName="M-Name7"
FirstName="FirstName" Title="Mr." NoPost="0" CreditLimit="-974" ProfileId="37655"
HotelId="1" />
  <PostListItem RoomNumber="350" ReservationId="33900" LastName="M-Name8"
FirstName="FirstName" Title="Mrs." NoPost="0" CreditLimit="-504" ProfileId="8023"
HotelId="1" />
  <PostListItem RoomNumber="370" ReservationId="34470" LastName="M-Name9"
FirstName="FirstName" Title="Mrs." NoPost="0" CreditLimit="100" ProfileId="38795"
HotelId="1" />
  <PostListItem RoomNumber="387" ReservationId="23116" LastName="M-Name10"
FirstName="FirstName" Title="Mr." NoPost="0" CreditLimit="-671.8" ProfileId="9372"
HotelId="1" />
  <PostListItem RoomNumber="398" ReservationId="35030" LastName="M-Name11"
FirstName="FirstName" Title="Mrs." NoPost="0" CreditLimit="296" ProfileId="39242"
HotelId="1" />
```

```

<PostListItem RoomNumber="399" ReservationId="27241" LastName="M-Name12"
FirstName="FirstName" Title="Mr." NoPost="0" CreditLimit="100" ProfileId="37655"
HotelId="1" />
<PostListItem RoomNumber="417" ReservationId="33464" LastName="M-Name13"
FirstName="FirstName" Title="Mrs." NoPost="0" CreditLimit="87.8" ProfileId="37856"
HotelId="1" />
<PostListItem RoomNumber="418" ReservationId="33705" LastName="M-Name14"
FirstName="FirstName" Title="Mr." NoPost="0" CreditLimit="-756.3"
ProfileId="37655" HotelId="1" />
<PostListItem RoomNumber="465" ReservationId="31046" LastName="M-Name15"
FirstName="FirstName" Title="Mrs." NoPost="0" CreditLimit="-314.8"
ProfileId="34887" HotelId="1" />
<PostListItem RoomNumber="585" ReservationId="33342" LastName="M-Name16"
FirstName="FirstName" Title="Mr. & Mrs." NoPost="0" CreditLimit="-2486.7"
ProfileId="37324" HotelId="1" />
</PostList>

```

**NOTE:** Characters over decimal 126 are ALWAYS sent as their respective entity. (LastName="Näme" would be sent as: LastName="N&#xE4;me")

## 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:

```

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

```

<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:

```

<PostList SequenceNumber="1236" HotelId="1" PaymentMethod="16" RevenueCenter="1"
WaiterId="Waiter1" WorkstationId="POS1" Date="070905" Time="194121">
<PostListItem RoomNumber="123" ReservationId="35316" LastName="Name"
FirstName="FirstName" Title="Mr." NoPost="0" CreditLimit="90.9" ProfileId="39527"
HotelId="1" />
</PostList>

```

## Inquiry for room number "217"

```

<PostInquiry InquiryInformation="217" MaximumReturnedMatches="16"
SequenceNumber="1241" 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'):

```
<PostList SequenceNumber="1241" HotelId="1" PaymentMethod="16"
RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" Date="070905"
Time="194121">
  <PostListItem RoomNumber="217" ReservationId="24331" LastName="Name1"
FirstName="FirstName" Title="Mrs." NoPost="0" CreditLimit="-236" ProfileId="27135"
HotelId="1" />
  <PostListItem RoomNumber="217" ReservationId="24332" LastName="Name2"
FirstName="FirstName" Title="Mr." NoPost="0" CreditLimit="1.8" ProfileId="27136"
HotelId="1" />
</PostList>
```

## Inquiry for not checked in RoomNumber

```
<PostInquiry InquiryInformation="444" MaximumReturnedMatches="16"
SequenceNumber="1243" RequestType="4" PaymentMethod="16" Date="070905"
Time="194121" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" />
```

Negative response:

```
<PostAnswer AnswerStatus="NG" ResponseText="Room not found"
SequenceNumber="1243" HotelId="1" Date="070905" Time="194121" PaymentMethod="16"
RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" />
```

### 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

```
<PostInquiry InquiryInformation="1040022102035774" MaximumReturnedMatches="16"
SequenceNumber="1250" RequestType="2" PaymentMethod="16" Date="070905"
Time="194121" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" />
```

Positive response:

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

### 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

```
<PostInquiry InquiryInformation="Name" SequenceNumber="17430" RequestType="12"
PaymentMethod="22" Date="121109" Time="122044" RevenueCenter="1"
WaiterId="WAITER2" WorkstationId="POS1" />
<PostList SequenceNumber="17430" HotelId="1|2" PaymentMethod="22"
RevenueCenter="1" WaiterId="WAITER2" WorkstationId="POS1" Date="121109"
Time="122052">
  <PostListItem RoomNumber="120" ReservationId="15761917" LastName="Name1"
FirstName="FirstName1" PaymentMethod="CA" NoPost="0" CreditLimit="990000000000"
ProfileId="117508" HotelId="1" />
  <PostListItem RoomNumber="437" ReservationId="25764460" LastName="Name2"
FirstName="FirstName2" PaymentMethod="CA" NoPost="0" CreditLimit="990000000000"
ProfileId="244044" HotelId="2" />
</PostList>
```

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

## POSTINGS

### Correct Posting

#### Inquiry

```
<PostInquiry InquiryInformation="D" MaximumReturnedMatches="16"
SequenceNumber="1280" RequestType="14" PaymentMethod="16" Date="070905"
Time="194121" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" />
PostList
<PostList SequenceNumber="1280" HotelId="1" PaymentMethod="16" RevenueCenter="1"
WaiterId="Waiter1" WorkstationId="POS1" Date="070905" Time="194121">
  <PostListItem RoomNumber="238" ReservationId="35746" LastName="DName1" NoPost="0"
CreditLimit="50" ProfileId="39767" HotelId="1" />
  <PostListItem RoomNumber="341" ReservationId="35797" LastName="DName2"
FirstName="FirstName" NoPost="0" CreditLimit="50" ProfileId="39785" HotelId="1" />
  <PostListItem RoomNumber="365" ReservationId="34865" LastName="DName3"
FirstName="FirstName" Title="Mr." Vip="001" NoPost="0" CreditLimit="-275.35"
ProfileId="39106" HotelId="1" />
  <PostListItem RoomNumber="464" ReservationId="31045" LastName="DName4"
FirstName="FirstName" Title="Mrs." NoPost="0" CreditLimit="-163" ProfileId="34886"
HotelId="1" />
  <PostListItem RoomNumber="9020" ReservationId="33772" LastName="DName5"
FirstName="FirstName" Title="Mr." NoPost="0" CreditLimit="1000237.85"
ProfileId="21046" HotelId="1" />
  <PostListItem RoomNumber="9006" ReservationId="33948" LastName="DName6"
FirstName="FirstName" 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 (DName1) can be referenced as `MatchfromPostList="1"`. The last one (DName6) from the list would be `MatchfromPostList="6"`.

An incorrect link leads to a denied posting.

## Postrequest for 'DName4', 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.

```
<PostRequest RoomNumber="464" ReservationId="31045" LastName="DName4"
HotelId="1" RequestType="4" InquiryInformation="D" MatchfromPostList="4"
SequenceNumber="1281" TotalAmount="12750" CreditlimitOverride="Y"
PaymentMethod="16" Covers="4" RevenueCenter="1" ServingTime="1"
CheckNumber="54586562" Subtotal1="8000" Subtotal2="3500" Discount1="-1500"
Tip="550" ServiceCharge1="2200" ServiceCharge2="0" Date="070906" Time="224700"
WaiterId="Waiter_6" WorkstationId="POS1" />
```

Positive response:

```
<PostAnswer RoomNumber="464" ReservationId="31045" LastName="DName4"
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 'DName4', 4th match of the above list but MatchfromPostList is set incorrectly

```
<PostRequest RoomNumber="464" ReservationId="31045" LastName="DName4" 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="DName4"
AnswerStatus="UR" ResponseText="Unprocessable request" CheckNumber="54586562"
SequenceNumber="1280" HotelId="1" Date="070906" Time="224700" PaymentMethod="16"
RevenueCenter="1" WaiterId="Waiter_6" WorkstationId="POS1" />
```

## PostingRequest containing POS Cash Rounding Difference amount

```
<PostRequest HotelId="1" RequestType="1" SequenceNumber="45335"
TotalAmount="16080" CreditlimitOverride="" PaymentMethod="1" Covers="1"
RevenueCenter="1" ServingTime="1" CheckNumber="12345678" Subtotal1="1550"
Subtotal2="2100" Subtotal3="250" Subtotal4="180" Subtotal5="2750" Subtotal6="650"
Subtotal7="1000" Subtotal8="150" Subtotal9="3500" Subtotal10="900"
Subtotal11="1350" Subtotal12="300" Subtotal13="780" Subtotal14="450"
Subtotal15="200" Subtotal16="950" Subtotal17="3" Tip="200" Discount1="-1283"
ServiceCharge1="100" Date="200427" Time="093942" WaiterId="Waiter_1"
WorkstationId="POS1" />
```

Total Amount	=	160.80 (rounded up to 0.05)
Subtotal1	=	15.50
Subtotal2	=	21.00
Subtotal3	=	2.50
Subtotal4	=	1.80
Subtotal5	=	27.50
Subtotal6	=	6.50
Subtotal7	=	10.00
Subtotal8	=	1.50
Subtotal9	=	35.00
Subtotal10	=	9.00
Subtotal11	=	13.50
Subtotal12	=	3.00
Subtotal13	=	7.80
Subtotal14	=	4.50
Subtotal15	=	2.00
Subtotal16	=	9.50
Tip	=	2.00
Discount1	=	-12.83
ServiceCharge1	=	1.00
Subtotal17	=	0.03 (POS Cash Rounding Difference amount)

Sum of Subtotals1-16, Tip, ServiceCharge1 and Discount1 = 160.77  
POS rounded the amount up to 160.80 as TotalAmount  
Subtotal17 is the rounding difference amount caused by the discount amount

## Posting, where the sum of all Subtotals does not match

```
<PostRequest RoomNumber="498" ReservationId="33265" LastName="Name" 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="Name"
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,ServiceCharges 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.

```
<PostInquiry InquiryInformation="1040022102035774" MaximumReturnedMatches="16"
SequenceNumber="1250" RequestType="2" PaymentMethod="16" Date="070905"
Time="194121" RevenueCenter="1" WaiterId="Waiter1" WorkstationId="POS1" />
```

Positive response:

```
<PostList SequenceNumber="1250" HotelId="1" PaymentMethod="16" RevenueCenter="1"
WaiterId="Waiter1" WorkstationId="POS1" Date="070905" Time="194121">
```

```
<PostListItem RoomNumber="223" ReservationId="34834" LastName="Name"
FirstName="FirstName" Title="Mr." NoPost="0" CreditLimit="-198.2"
ProfileId="39077" HotelId="1" />
```

```

</PostList>

<PostRequest RoomNumber="223" ReservationId="34834" LastName="Name" HotelId="1"
RequestType="4" InquiryInformation="1040022102035774" MatchfromPostList="1"
SequenceNumber="1251" TotalAmount="4600" CreditlimitOverride="" PaymentMethod="16"
Covers="1" RevenueCenter="1" ServingTime="1" CheckNumber="12345678"
Subtotal1="1000" Subtotal2="1000" Subtotal3="1000" Subtotal4="1000" Discount1=""
Discount2="" Discount3="" Discount4="" Tip="200" ServiceChargel="100"
ServiceCharge2="100" ServiceCharge3="100" ServiceCharge4="100" Tax1="160"
Tax2="160" Tax3="160" Tax4="160" Date="070905" Time="194122" WaiterId="Waiter_1"
WorkstationId="POS1" />
<PostAnswer RoomNumber="223" ReservationId="34834" LastName="Name"
AnswerStatus="OK" ResponseText="Request successful" CheckNumber="12345678"
SequenceNumber="1251" HotelId="1" Date="070905" Time="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.

```

<PostRequest RoomNumber="" ReservationId="" LastName="" HotelId="1"
RequestType="1" InquiryInformation="" MatchfromPostList="" SequenceNumber="1290"
TotalAmount="12750" CreditlimitOverride="N" PaymentMethod="35" Covers="4"
RevenueCenter="1" ServingTime="1" CheckNumber="54586562" Subtotal1="8000"
Subtotal2="3500" Discount1="-1500" Tip="550" ServiceChargel="2200" Date="070906"
Time="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."

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 RoomNumber="9400" ReservationId="28633" LastName="POS Cash Account"
AnswerStatus="OK" ResponseText="Request successful" CheckNumber="54586562"
SequenceNumber="1290" HotelId="1" Date="070906" Time="224700" PaymentMethod="35"
RevenueCenter="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="DName1" NoPost="0"
CreditLimit="50" ProfileId="39767" HotelId="1" />
  <PostListItem RoomNumber="341" ReservationId="35797" LastName="DName2"
FirstName="FirstName" NoPost="0" CreditLimit="50" ProfileId="39785" HotelId="1" />
  <PostListItem RoomNumber="365" ReservationId="34865" LastName="DName3"
FirstName="FirstName" Title="Mr." Vip="001" NoPost="0" CreditLimit="-275.35"
ProfileId="39106" HotelId="1" />
  <PostListItem RoomNumber="464" ReservationId="31045" LastName="DName4"
FirstName="FirstName" Title="Mrs." NoPost="0" CreditLimit="-163" ProfileId="34886"
HotelId="1" />
  <PostListItem RoomNumber="9020" ReservationId="33772" LastName="DName5"
FirstName="FirstName" Title="Mr." NoPost="0" CreditLimit="1000237.85"
ProfileId="21046" HotelId="1" />
  <PostListItem RoomNumber="9006" ReservationId="33948" LastName="DName6"
FirstName="FirstName" 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 'DName4', 4th match of the above list to void this transaction

```
<PostRequest RoomNumber="464" ReservationId="31045" LastName="DName4" 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" Discount1="1500" Tip="-550" ServiceCharge1="-
2200" Date="070905" 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.

```
<PostRequest RoomNumber="" ReservationId="" 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" Discount1="1500" Tip="-550" ServiceCharge1="-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.

# 5 Appendix A – Code Tables

## AnswerStatus

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

"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.

Value	Meaning
1	Posting of Cash/Creditcard (PostRequest only)
2	Inquiry by Track2 of a magnetic stripe
4	Inquiry by RoomNumber
8	Inquiry by guest's name

- 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"**.