

# Oracle® Hospitality Symphony

## Enhanced PMS Interface Specifications



Release 2.9 and Later  
F16307-01  
February 2019

ORACLE®

Oracle Hospitality Symphony Enhanced PMS Interface Specifications, Release 2.9 and Later

F16307-01

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 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  | iv         |
| <hr/>  |            |
| <b>1 Message Formats and Interface Methods</b> | <b>1-1</b> |
| <hr/>  |            |
| Interface Methods                              | 1-1        |
| Message Format                                 | 1-1        |
| Pinging  | 1-4        |
| <br>   |            |
| <b>2 Application Data Field Types</b>          | <b>2-1</b> |
| <hr/>  |            |
| Nx   | 2-1        |
| \$x  | 2-1        |
| Ax   | 2-1        |
| <br>   |            |
| <b>3 Application Data Format</b>               | <b>3-1</b> |
| <hr/>  |            |
| Charge Posting                                 | 3-1        |
| Guest Charge Posting Dialog                    | 3-4        |
| Non Front-Office Charge Posting Dialog         | 3-4        |
| Service Charge Itemizer Details                | 3-5        |
| Tax Handling                                   | 3-5        |
| Check Facsimile                                | 3-6        |

---

# Preface

## Purpose

This document contains information about designing an interface between Oracle Hospitality Symphony and an Enhanced Property Management System (PMS).

## Audience

The audience for this document is:

- Programmers who design software interfaces between Symphony POS systems and property management systems
- Field engineers and technicians who troubleshoot problems with interfaces between Symphony POS systems and property management systems

The following prerequisites are necessary for the reader:

- Clear understanding of the charge posting specifications of the PMS interface being programmed, and a thorough understanding of how to implement the interface
- Understanding of how to program the specific product involved to enable the charge posting interface

## 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 and any associated log files
- Screenshots 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/>

## Revision History

---

| Date          | Description of Change   |
|---------------|---|
| February 2019 | <ul style="list-style-type: none"><li>• Initial Publication</li><li>• Conversion to correct template format</li></ul> |

---

# 1 Message Formats and Interface Methods

To develop the third-party PMS Interface, use the message formats and the interface methods described in this section.

## Interface Methods

The third-party PMS Interface supports a TCP-based interface.

### TCP Interface

This interface connects Symphony to Windows®-based systems and other systems using the TCP/IP networking protocol. This interface facilitates communications between the POS application and third-party applications that reside on the same Windows platform as the Symphony software.

This interface is also compatible with many forms of local area networks (LANs), including Ethernet, etc.

#### TCP Connection

The Symphony POS System connects to the TCP port as a socket-client. The third-party PMS application interface must act as a socket-server and accept TCP connections from the Symphony POS System on the default port 5009. This port should be configurable so that changes to any value above 5009 if needed. The Symphony POS System will not close the port unless it is shutting down, but includes of course functionality to determine if the third-party PMS application disconnected and will reconnect when needed. The third party PMS application interface on the other side should not close the port unless it is shutting down but should include functionality to determine if the Symphony POS System client disconnected.

Even though a Symphony POS System can handle up to hundreds of workstations there will only be one interface connection (i.e. open socket-pair) to the third-party PMS application per physical Symphony server.

## Message Format

The format of all messages between the Symphony POS System and the third-party PMS application include three segments and have the following layout:

**SOH** <POS Source ID> **STX FS** <Application Seq.> <Application Data> **ETX**  
<Checksum> **EOT**

## SOH

The SOH character (Start of Header) serves as a message lead-in character that identifies the start of a new message. The SOH character is represented by the 7-bit hexadecimal value 01h, plus a parity bit, if applicable.

### <POS Source ID>

**SOH** <POS Source ID> **STX FS** <Application Seq.> <Application Data> **ETX**  
<Checksum> **EOT**

The <POS Source ID> character field identifies the source (when sent by the Symphony POS System) and the destination of a message (when replied by the third-party PMS application). The <POS Source ID> field contains either 18 or 25 characters. The interface definition record may be configured to supply either a two-character or a nine-character Workstation Number, which is followed by the Outgoing Message Name (16 characters). The Workstation Number is a 2 or 9 bytes numeric character field in the range 1 through 99 (or 1 to 999999999 respectively). It is represented by ASCII character codes in the range Hex 30 through Hex 39, plus parity bits, if applicable. The digits are right justified within the field with leading '0' (ASCII zero character, Hex 30) or leading blanks (ASCII blank character, Hex 20). Leading characters appear only if the number of characters in the field is less than the total field size. The Outgoing Message Name is a 16-byte alphanumeric character field stored in the Symphony Interface Definition. This field is represented by ASCII character codes, plus parity bits, if applicable. It identifies and differentiates the source(s) of the communications message(s).

Examples, the character "." represents a single SPACE (20h):

2 character Workstation Number:

01PMS INTERFACE...

9 character Workstation Number:

000000001PMS INTERFACE...

## STX

The STX (Start of TeXt) serves as a data field lead-in character that identifies the start of the message data block. The STX character is represented by the 7-bit hexadecimal value 02h, plus a parity bit if applicable.

## FS

The FS (Field Separator) identifies this message as a SIM message data block. The FS character is represented by the 7-bit hexadecimal value 1Ch, plus a parity bit if applicable.

### <Application Seq.>

**SOH** <POS Source ID> **STX FS** <Application Seq.> <Application Data> **ETX**  
<Checksum> **EOT**

The <Application\_Seq.> segment comprises a two-digit sequence number and a retransmission flag. Each Symphony POS workstation application increments its own sequence number with each message. When a message is retransmitted, the same sequence number will be used as the original message. In addition, a retransmission flag character is provided. The format of this segment is as follows:

| Field                         | Length  | Format         | Remarks   |
|-------------------------------|---------|----------------|---|
| Applications Sequence Number  | 2 Bytes | 2 ASCII digits | May contain leading spaces or zeroes and is between 01 and 99                         |
| Retransmission Flag Character | 1 Byte  | Space or "R"   | "R" character (ASCII 52h) is placed in this field if this is a retransmission message |

The Application\_Sequence number is initially set to "01" when the Symphony POS workstation applications starts. The application rolls the sequence number back to "01" after "99". If the third-party system receives a message containing the same sequence number as the previous message and the retransmission flag is set, the third party should retransmit the last response.

## <Application Data>

**SOH** <POS Source ID> **STX FS** <Application Seq.> **<Application Data>** **ETX**  
 <Checksum> **EOT**

The <Application Data> segment is the actual payload of the message. It leads with a message ID field that identifies the message type. Permissible message IDs are listed on the following pages. A variable number of fields follow the message ID field. These fields constitute the body of the actual message and are defined for each message type. These fields are variable in length, consequently, each field will be separated by an ASCII File Separator character (FS = 1Ch) which serves to determine the start of each field. The format of applications data segment looks like this:

<Message ID> **FS** <Msg Field 1> **FS** <Msg Field 2> **FS** .... **FS** <Msg Field n>

The total size of the message can be 32K from the SOH to the EOT. There are a maximum of 37 bytes overhead, which means that the maximum byte count of all fields and field separators is 32768 – 37, or 32731 bytes.

## <Checksum>

**SOH** <POS Source ID> **STX FS** <Application Seq.> <Application Data> **ETX**  
 <Checksum> **EOT**

The Checksum field is only used when communicating over an asynchronous serial interface. A TCP-based PMS Interface will ignore this field, so it can be omitted from the

message format. When the Checksum is part of the message, however, format it as follows:

| Field    | Length  | Format                 | Remarks  |
|----------|---------|------------------------|--|
| Checksum | 4 bytes | 4 ASCII Hex characters | Contains ASCII characters in the range 30H-39H and 41H-46H (0-9 and A-F) |

The Checksum is the 16-bit binary addition (excluding parity, if applicable) of all characters after the SOH, up to and including the ETX character. The Checksum is initially set to zero. For asynchronous, serial transmission, the Checksum is represented as four ASCII-Hex characters.

## EOT

The EOT (End Of Transmission) serves as a message lead-out character and identifies the end of a message. The EOT character is represented by the 7-bit hexadecimal value 04h, plus a parity bit, if applicable.

## Pinging

The TCP connection has a typical “keep-alive” time-out of two hours. In order to detect a “down” interface more quickly and re-establish the connection, the Symphony PMS interface periodically sends a “ping” message to the server about every five minutes. The third-party PMS application should detect the ping message and return the message in its original format. The format of the ping message should be as follows:

**SOH <POS Source ID> STX ETX EOT**

The <POS Source ID> segment contains a null address: the workstation number will be zero and the interface name will be blank.

Example, the numbers in square brackets “[ ]” represent the decimal ASCII value of a single, non-printable character and the character “.” represents a single SPACE (20h):

2 character Workstation Number:

[1]·0··········[2][3][4]

9 character Workstation Number:

[1]········0··········[2][3][4]



## 2 Application Data Field Types

Application data field types for both request and response messages are detailed in terms of field description, field size, and field format. The field description identifies the way in which the data is interpreted (e.g., employee number, revenue center number, etc.). The field size indicates the maximum number of characters that are in the given field. The field format describes the field in terms of data type.

The following is a description of the five basic field format data types:

### Nx

Unsigned numeric character field (without decimal point). The ASCII characters 0 through 9 represent all numeric characters. The value, x, defines the number of digits possible in the field. Padding (i.e. leading blanks) is not used.

### \$x

Signed numeric character field (with decimal point). The ASCII characters 0 through 9 represent all numeric characters. The value, x, defines the number of digits possible in the field, including the decimal point. Depending on the configuration in Symphony the decimal point character is either '.' (ASCII character, 2Eh) or ',' (ASCII character, 2Ch). Padding (i.e. leading blanks) is not used.

The negative sign precedes the most significant digit and is represented by an ASCII "-" character, 2Dh for negative numbers.

### Ax

Alphanumeric character string field. All characters represented by the ASCII characters in the range Hex 0x20 through Hex 0xFF. International characters (from Hex 0x80) depend on the configuration of the Symphony POS system.

The value, x, defines the maximum number of characters possible in the field. Padding (i.e. leading or trailing blanks) is not used.

# 3 Application Data Format

## Charge Posting

This message pair is used by the Symphony POS system to perform a charge posting. Leading message ID fields, that identifies the message type, are shown in blue, bold font and are not included in the related tables that explain the other fields.

### Charge Posting Request

Symphony POS to third-party PMS Interface Application Data Format:

There are two formats to initiate a Charge Posting:

- a) Posting a non Front-Office charge (i.e. Cash)

**IFC\_CHG\_PST**, **FS**, SPACE, **FS**, SPACE, **FS**, Tender\_num, **FS**, Tender\_amount, **FS**, Num\_sales\_itemizers, **FS**, Sales Itemizer 1,..., Sales Itemizer N, **FS**, Num\_discounts, **FS**, Discount 1,...,Discount n, **FS**, Num\_service\_charges, **FS**, Service charge 1,..., Service charge N, **FS**, Num\_taxes, **FS**, Tax Itemizer 1, ...,Tax Itemizer N, **FS**, Guest Check Number, **FS**, Check Employee Number, **FS**, Serving Period/Cashier Number, **FS**, Revenue Center Number, **FS**, Number of Guests, **FS**, Date, **FS**, Time, **FS**, Previous Payment Amount

SPACE = hexadecimal value 20h, details of each field can be found on the following pages.

The response to this posting is found below at “Acknowledging a charge posting”.

- b) Selecting a guest to whom to post a charge:

**IFC\_CHG\_PST**, **FS**, Guest\_id, **FS**, Tender\_num

| Field Name | Format | Description   |
|------------|--------|---|
| Guest_id   | A30    | Guest ID as entered by POS operator. This is not limited other than in input length. It is up to the third-party PMS on how this information should be handled to source a particular guest or a list of guests |
| Tender_num | N9     | Symphony Tender/Media Number that is used for the Charge Posting.   |

## Charge Posting Response

The third-party PMS will answer with a message type IFC\_GST\_SEL and the operator will select/confirm the Guest. Once done an IFC\_CHG\_PST Charge Posting will be send, including the selection details as well as the posting details.

Returning a list of guests from which to select:

**IFC\_GST\_SEL**, **FS**, Guest\_id, **FS**, IFC\_CHG\_PST, **FS**, List\_size, **FS**, Guest\_list[]

This message is returned when a IFC\_CHG\_PST message is sent requesting a guest selection.

| Field Name  | Format | Description   |
|-------------|--------|---|
| Guest_id    | A30    | Guest ID AS entered by POS operator: Should echo what was send in the Charge Posting request.     |
| IFC_CHG_PST | A11    | Hard-Coded Value set to 'IFC_CHG_PST'   |
| List_size   | N5     | Length of guest list  |
| Guest_list  | A40    | List of guest names from which to select for a charge posting, each name separated with <b>FS</b> |

## Charge Posting Request for a Selected Guest

**IFC\_CHG\_PST**, **FS**, Guest\_id, **FS**, Guest\_name, **FS**, Sel\_num, **FS**, Tender\_num, **FS**, Tender\_amount, **FS**, Num\_sales\_itemizers, **FS**, Sales Itemizer 1,..., Sales Itemizer N, **FS**, Num\_discounts, **FS**, Discount 1,...,Discount N, **FS**, Num\_service\_charges, **FS**, Service charge 1,..., Service charge N, **FS**, Num\_taxes, **FS**, Tax Itemizer 1, ..,Tax Itemizer N, **FS**, Guest Check Number, **FS**, Check Employee Number, **FS**, Serving Period/Cashier Number, **FS**, Revenue Center Number, **FS**, Number of Guests, **FS**, Date, **FS**, Time, **FS**, Previous Payment Amount

| Field Name          | Format | Description  |
|---------------------|--------|--|
| Guest_id            | A30    | Guest ID AS entered by POS operator: Should echo what was send in the Charge Posting request.  |
| Guest_name          | A40    | Name of selected guest   |
| Selection_num       | A6     | 'S' plus the index of the selected guest (based on Guest_list returned from PMS in IFC_GST_SEL)  |
| Tender_num          | N9     | Simphony Tender/Media Number that is used for the Charge Posting.  |
| Tender_amount       | \$12   | The tender or payment amount   |
| Num_sales_itemizers | N5     | The number of sale itemizers included in the message. Typically 16 for Simphony.   |
| Sales Itemizer 1..N | \$12   | Sales Itemizers 1 through n, each separated with <b>FS</b>   |
| Num_discounts       | N5     | The number of discounts itemizers in the message, the maximum is 16. Discounts itemizers reflect any discount applied to the related Sales Itemizer. |
| Discount 1..N       | \$12   | Discount Itemizers 1 through n, each separated with <b>FS</b>  |

|  |      |   |
|--|------|---|
| Num_service_charges                      | N5   | The number of service charge itemizers in this message. The maximum is 14   |
| Service Charge 1..N                      | \$12 | Service Charge Itemizers 1 through n, each separated with <b>FS</b>   |
| Num_taxes                                | N5   | Number of tax itemizers in the message. The maximum is 63.  |
| Tax Itemizer 1 ..N                       | \$12 | Add-On Tax Collected or Taxable Total (for Inclusive taxes, i.e. Vat) Itemizer 1 through n, each separated with <b>FS</b> |
| Guest Check Number                       | N8   | Guest check number assigned to the guest check in question.   |
| Check Employee Number                    | N9   | Guest check employee number   |
| Serving Period/Cashier/Order-Type Number | N9   | Serving period, cashier or order-type number, depending on the setup of the interface used for this posting.              |
| Revenue Center Number                    | N5   | Revenue center number used for this posting   |
| Number of Guests                         | N5   | Number of guests assigned to the guest check.   |
| Date                                     | A8   | Date the check was begun in the format YYYYMMDD   |
| Time                                     | A8   | Time the check was begun in the format HH:MM:SS   |
| Previous Payment Amount                  | \$12 | Any previous payment amount already applied to this guest check.  |

## Charge Posting Acknowledgement Response

**IFC\_CHG\_PST**, **FS**, Message\_Status, **FS**, Status, **FS**, Message, **FS**, TndTTL

| Field Name     | Format | Description   |
|----------------|--------|---|
| Message_Status | A1     | 'Y' if a guest message is pending, 'N' to bypass any guest message functionality  |
| Status         | A1     | Charge posting status using one of the possible values below:<br>'P' = Posted<br>'D' = Declined<br>'E' = Error  |
| Message        | A30    | If Status 'P' -- Message is placed in transaction detail and usually contains room# and guest-name.<br>If Status 'D' or 'E' – Message is displayed to indicate problem to the operator. |
| TndTTL         | \$12   | Not used  |

## Guest Charge Posting Dialog

The typical posting dialog is (FS not listed for simplicity):

| Simphony POS   |   | Third-party PMS  |
|--|---|--|
| <b>IFC_CHG_PST</b> , Guest_id,<br>Tender_num                 | ⇒ |  |
|  | ⇐ | <b>IFC_GST_SEL</b> , Guest_id,<br>IFC_CHG_POS, List_size, Guest_list[] |
| POS Operator selects the appropriate guest                   |   |  |
| <b>IFC_CHG_PST</b> , Guest_id,<br>Guest_name, Sel_num, ..... | ⇒ |  |
|  | ⇐ | <b>IFC_CHG_PST</b> , message_status, status,<br>message, .....         |

## Non Front-Office Charge Posting Dialog

The typical posting dialog is (FS not listed for simplicity):

| Simphony POS  |   | Third-party PMS  |
|---|---|--|
| <b>IFC_CHG_PST</b> , SPACE, SPACE,<br>Tender_num, ..... | ⇒ |  |
|   | ⇐ | <b>IFC_CHG_PST</b> , message_status, status,<br>message, ..... |

## Service Charge Itemizer Details

Depending on the Symphony POS setup, the following breakdowns are provided:

### Full Breakdown of Service Charges with 14 Itemizers

| Itemizer |   |                               |
|----------|---|-------------------------------|
| 1        | Service-Charge based on SVC Itemizer 1                    | See remark (1) below          |
| 2        | Service-Charge based on SVC Itemizer 2                    | See remark (1) below          |
| 3        | Service-Charge based on SVC Itemizer 3                    | See remark (1) below          |
| 4        | Service-Charge based on SVC Itemizer 4                    | See remark (1) below          |
| 5        | Service-Charge based on SVC Itemizer 5                    | See remark (1) below          |
| 6        | Service-Charge based on SVC Itemizer 6                    | See remark (1) below          |
| 7        | Service-Charge based on SVC Itemizer 7                    | See remark (1) below          |
| 8        | Service-Charge based on SVC Itemizer 8                    | See remark (1) below          |
| 9        | OPEN Amount Service-Charge with no SVC Itemizer enabled   | See remark (2) below          |
| 10       | PRESET Amount Service-Charge with no SVC Itemizer enabled | See remark (2) below          |
| 11       | Stored Value Transaction Service Charge                   | Simphony SVC Option Bit 14 ON |
| 12       | Non-Revenue Service Charge                                | Simphony SVC Option Bit 7 ON  |
| 13       | Charged Tip   |                               |
| 14       | Auto Service-Charge                                       |                               |

Remark (1): Amount of any PERCENTAGE based Service Charges based on the SVC Itemizer (1-8) and the Menu-Items present on the check. For an AMOUNT based Service Charge based on the FIRST active SVC Itemizer found (if any).

Remark (2): Only used with AMOUNT based Service Charges with no SVC Itemizer enabled.

### Limited Breakdown of Service Charges with 2 Itemizers

| Itemizer |  |
|----------|--|
| 1        | Sum of amount and percentage based Service-Charges |
| 2        | Auto Service-Charge                                |

## Tax Handling

Simphony POS differs between two basic Tax configurations, an Add-On Tax option that covers both Add-On and Inclusive taxes combined in a single transaction and a VAT Tax option that will only permit inclusive, percentage based taxes.

Depending on this Tax configuration, the Tax Itemizer (1 to 8 or 1 to 64) will contain one of the following:

- Add-On tax collected only when Symphony POS is configured to use the Add-On Tax option. Any inclusive, percentage based taxes used in such a configuration will not be included.

- Taxable Totals when Symphony POS is configured to use the VAT Tax option.

## Check Facsimile

This message pair is used by the Symphony POS system to provide the check facsimile to the third-party PMS. The check facsimile is a textual representation of the printed POS receipt without any formatting instructions.

### Check Facsimile Request

**FO\_FI\_FINAL\_TENDER**, **FS**, Date, **FS**, Time, **FS**, Revenue Center Number, **FS**, Guest Check Number, **FS**, Chk\_Data

| Field Name            | Format       | Description  |
|-----------------------|--------------|--|
| Date                  | A8           | Date when the check was begun in the format YYYYMMDD   |
| Time                  | A8           | Time when the check was begun in the format HH:MM:SS   |
| Revenue Center Number | N5           | Revenue Center of this check   |
| Guest Check Number    | N8           | Guest check number assigned to the guest check in question.  |
| Chk_Data              | A<see desc.> | The actual check facsimile, variable in length. Each line is terminated with LF (0Ah). Each line can have a maximum length of 40 characters, the total size of this field on the other side can range up to 32k. |

### Check Facsimile Response

**FO\_87\_DONE**, **FS**, Text

| Field Name | Format | Description   |
|------------|--------|---|
| Text       | A70    | Up to the PMS to fill this Field. It is neither displayed nor logged on the POS side. |