

Oracle® Hospitality Payment Interface Opera Web Service (OWS) OPI Installation Guide



Release 20.2
November 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

Contents	3
Preface	4
1 Opera Web Service (OWS)	1-1
2 OWS vs. KIOSK	2-1
OWS	2-1
Kiosk	2-1
3 Introduction	3-1
4 Dataflow	4-1
5 Communication Flow Diagram & Use Cases	5-1
6 Kiosk Install for V5 Environments	6-1
O2G Kiosk Interface Setup Form	6-3
OPERA Sample Screenshots	6-6
7 Kiosk Install for 1.20 Environments	7-1
1.20 Environment - License Activation	7-1
1.20 Environment - Functionality Setup	7-2
8 Kiosk Install for OPERA Cloud 20.X Environments	8-1
OPERA Cloud 20.X - License Activation	8-1
OPERA Cloud 20.X - Functionality Setup	8-2
9 FAQ & Sample XML Messages	9-1
OWS messages for Kiosk use	9-1

Preface

Purpose

This document describes how to organize environments for an installation of the Oracle Payment Interface (OPI) for OPERA Web Service (OWS) On Premise Token Exchange Service

Audience

This document is intended to cover the steps required to set up OPI for Opera Web Service to handle the On Premise and Hosted Token Exchange functionality.

This document covers only the configuration of the OWS functionality, it does not cover in detail, installation of the OPI software and IFC8 merchant configuration, separate documentation already exists to cover this.

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
September 2020	<ul style="list-style-type: none">• Initial Document Creation

1

Opera Web Service (OWS)

OWS is a collection of Windows based Web Services that provide access to OPERA Functionality, and can also act as an interface between external applications such as Web Booking Engines, and Kiosks.

Any reservation from OWS with a Primary Account Number(PAN) is sent to the Token Proxy Service (TPS) to tokenize and then the token is stored with the reservation in Property Management System(PMS).

Any reservation from OWS sent with a Token, can be sent directly through to PMS.

2

OWS vs. KIOSK

OWS

- An OPERA Electronic Distribution Suite(OEDS) installation with its own OPW_500 license.
- Scheduled as a 24 hour installation as it requires some Opera configuration and training (8 less hours for hosted).
- OWS will not always have a vendor to schedule, as it is commonly used by the site's own website's booking engine and therefore may be tested by the hotel themselves (This would be considered a 'Generic OWS').
- Rare Medium, Sandrix, and Owner Relations are examples of OWS vendors you may encounter.

Kiosk

- An OEDS installation that has its own PMS Add-On license OPP_KSK.
- It does not require an OWS license to operate and therefore an OWS license does not need to be sold along with the OPP_KSK license.
- Utilizes OWS services on the OEDS server, which is why it is commonly associated with OWS (Currently there are no specific KIOSK services installed on the OEDS server, however there were in the past).

Not all Kiosk interfaces follow this rule. Intelity (KeyPR) is a popular kiosk interface that uses Kiosk, OWS, and even OXI, if used at full potential. Depending on what the site wants, Intelity would require each license purchased individually and would require a minimum of 3 days (less 8 hours for hosted). Always refer to the Kiosk Vendor installation guide's first few pages to confirm required licenses.

NOTE:

For OPI/KIOSK integration, O2G must be configured in the Channel Configuration, which does require an OWS license to activate. However this is NOT considered the same thing as purchasing the OWS license and therefore would not be billed to the client as such.

3

Introduction

When OPI is connected to OPERA, the Kiosk vendor has the ability to activate a pin-pad device just like a front desk workstation. This ensures the authorization and settlement functionality remains the same in all areas of the application. The following is needed for the solution to work correctly:

- OPERA PMS connected to a validated Payment Service Provider (PSP) via OPI.
- OWS-KIOSK is installed on the OEDS server.
- The OPERA Kiosk license code must be activated.
- The OPERA2Go license code must be activated.
- The Payment Terminals are configured in OPERA Configuration - Valid SSL Certificate is active on OEDS server.

4

Dataflow

The following describes the dataflow the vendor follows for integration to OWS-KIOSK with a connected OPI interface:

- The Vendor performs **Reservation.FetchBooking** to retrieve the reservation from OPERA PMS.
- The Vendor performs **Information.QueryLov** to retrieve the list of configured devices. This is an optional request if the vendor has hardcoded the device list on their side. Note the values returned will either be the encoder value in OPI or if the OPI vendor is set to middleware, then the terminal ID is set in the middleware.
- The Vendor performs **ResvAdvanced.AddPayment** to activate the device to put a new card into OPERA. This call sends a request to authorize the card.
- Check-out uses the card on file that was authorized at check-in unless a new card is provided.
- The Vendor, using a **ResvAdvanced.CheckIn** or **ResAdvanced.CheckOut**, will replace the card on file.

5

Communication Flow Diagram & Use Cases

Figure 5-1 OWS Contactless Check In

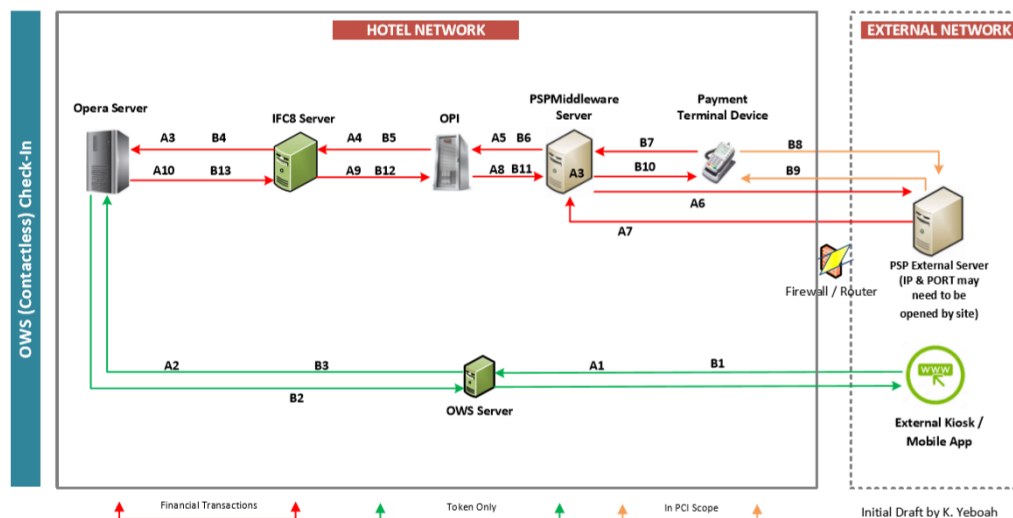


Figure 5-2 Check in Details

A. Check-In for Guests using Previously Acquired Token (Card NOT Present via Mobile App)

1. Card data sent from CRS to OWS/OXI
2. OWS/OXI sends Reservation data to Opera
3. OPERA Sends request to IFC8
4. IFC8 Sends request to OPI Server
5. OPI forwards reservation + authorization request PSP Middleware
6. PSP Middleware sends reservation + authorization request + Previously acquired Token to PSP External Server
7. PSP External Server returns the authorization confirmation to PSP Middleware
8. PSP Middleware forwards confirmation to OPI
9. OPI Sends confirmation to IFC8
10. IFC8 Completes request by sending confirmation to OPERA

B. Check-In for Guests using (Card Present via External Kiosk for e.g.)

1. Kiosk requests reservation details from Opera
2. Opera responds with Reservation data back to Kiosk
3. Kiosk sends add payment call (to activate Payment Terminal Device) to Opera via OWS
4. OPERA Sends request to IFC8
5. IFC8 Sends request to OPI Server
6. OPI forwards reservation + authorization request PSP Middleware
7. PSP Middleware sends reservation + authorization request to Payment Terminal Device (at Kiosk for e.g.)
8. Card is inserted into Payment PSP and information sent to PSP External Server
9. PSP External Server returns the new Token + authorization confirmation to Payment Terminal
10. Payment Terminal sends confirmation to PSP Middleware
11. PSP Middleware forwards token + confirmation to OPI
12. OPI Sends Token + confirmation to IFC8
13. IFC8 Completes request by sending confirmation to OPERA

6

Kiosk Install for V5 Environments

1. **Configuration > Setup> License codes > PMS> Add-On License .**
2. Scroll Down to the OPP_KSK record and click **EDIT** .
3. Enter License Code and place a tick into the **Active** Box to enable the interface in OPERA.
4. Select **OK**. If installing in a multi-property PMS environment, repeat for EACH PMS property.

Figure 6-1 Edit Addon License

- a. Select '**Generate Data**'
- b. Check '**Allow Check In**'
- c. Check '**Allow Check Out**'
- d. Select All Windows for CheckOut
- e. Select All Methods of Payments
5. If installing in a multi-property PMS environment, repeat for EACH PMS property.
6. Complete the rest of the setup via V5 configuration screens as noted in the manual:
 - a. **Setup > Kiosk Configuration > KIOSK Channel**
 - b. **Setup > Kiosk Configuration > O2G Channel**

- c. **Application Settings > OWS group > Settings > Fetch_Reservation** parameter to ALL (may require activation of the OWS license to change, then deactivate the OWS license).

7. . If the O2G is to be used, the following must set up:

- a. **OPERA Configuration> Interface Configuration>** Select the **OPI Interface >** Select **Workstation Setup** tab.

Figure 6-2 Workstation Setup Edit

Terminal	Encoder #	Location	Type
CHPPINDVC1	1	Chip & Pin Device 1	M
CHPPINDVC2	2	Chip & Pin Device 2	M

Figure 6-3 Workstation Setup Edit

Terminal	Encoder #	Location	Type
CHPPINDVC1	1	Chip & Pin Device 1	M
CHPPINDVC2	2	Chip & Pin Device 2	M

- b. Terminal –Maps to the encoder on the PSP side of OPI (if Middleware is used for the OPI vendor, then it needs to be same terminal ID as set in middleware). Do not select from the drop-down, type it in. **The Terminal is limited to 16 characters just like front desk setup.**
- c. Encoder# - Is not used, however may be required to enter. It can be the same as the Terminal, however due to [BUG 31108396](#), the value must be 10 characters or less. It is advised to use a numerical scheme, starting at 1 and going up on subsequent devices.
- d. Location – Free text
- e. Type – Must be sent to MOBILE ENCODER



NOTE:

If installing in a multi-property PMS environment, repeat for EACH PMS property.

O2G Kiosk Interface Setup Form

1. Select **Configuration> Setup> Kiosk Interface > Select O2G** from the Channel LOV and click **OK**
2. Set up the form as show here:
3. Select '**Generate Data**'
4. Check '**Allow Check In**'
5. Check '**Allow Check Out**'
6. Select All Windows for CheckOut
7. Select All Methods of Payments

If installing in a multi-property PMS environment, repeat these steps for EACH PMS property.

Figure 6-4 Kiosk Interface

Kiosk Interface

Guest Stationery

☐ Print

☐ Generate Files

☒ Generate Data

Room Type Change

☐ Allow Rate Change

☐ Allow if multiple rates

Price Delta %: - +

☒ Allow Check In

☒ Allow Check Out

☐ Checkout if all Windows paid by Guest

☐ Print only those windows paid by guest

☐ Credit Card Authorization by External System

☐ Generate Advance Bill at Check In

UDF Mappings

Windows for Check Out

Walkin Rate Codes

Default Market Code

Default Source Code

Available Package Codes

Method Of Payments

Check In Start Time Stop Time

Check Out Start Time Stop Time

02G

OK Close

Configuring Vendor OPI Devices in OPERA

This setup allows the interface to pull a list of encoders to choose from. Some vendors have encoders hardcoded on their side, other vendors provide the ability to select from list of devices.

1. To configure in OPERA> Select Configuration> Interface Configuration>
2. Select the OPI Interface > Select Workstation Setup tab>

Figure 6-5 Configuring Vendor OPI Devices

Terminal	Encoder #	Location	Type
MOBCONCIERGE	MOBCONCI	Mobile Concierge	M
MOBILE1	MOBILE1	Mobile 1	M
MOBILE2	MOBILE2	Mobile 2	M
MOBVALET	MOBVALET	Mobile Valet	M

Set up the form as shown here:

- a. Terminal – The Terminal maps to the encoder in OPI(if Middleware is used for the OPI vendor, then it **needs to be the same as the terminal ID** as set in middleware). Do not select it from a drop down, type it in manually. If the Terminal is longer than 10 characters, only use **the last 10**.
- b. Encoder# -
- c. Location – Free text
- d. Type – Must be sent to MOBILE ENCODER
- e. If installing in a multi-property PMS environment, repeat for EACH PMS property.

Figure 6-6 Workstation Setup Edit

NOTE:

Encoder# field must be kept to 10 characters or less for ALL OPI involving OWS.

OPERA Sample Screenshots

Check in With Authorization

Figure 6-7 Check In Authorization

Figure 6-8 User Activity Log Details

User	Time	Date	Action Type	Description
OEDSOWS	10:16	11/08/18		
OEDSOWS	10:16	11/08/18		
OEDSOWS	10:15	11/08/18		
SUPERVISOR	10:11	11/08/18		

User	OEDSOWS
Time	10:16
Date	11/08/18
Station ID	OPERA-IFC
Action Type	RECORD CREDIT CARD APPROVAL
Description	APPROVED 184.05 CAD, FOR PAYMENT TYPE MC, APPROVAL CODE: OK102Z, FOR CONFIRMATION #32650, FOR Smith-Schuster, Juju

Figure 6-9 Authorization Fields

<div><input checked="" type="checkbox"/> Authorizations</div> <div><input type="checkbox"/> Payments</div> <div><input type="checkbox"/> Refunds</div> <div><input type="checkbox"/> Failures</div>							
Date	Time	Amount	Approval	CC Type	Credit Card	Expiration	Transaction Type
11/22/17	10:16	184.05	OK102Z	MC	XXXXXXXXXXXXXXXX0009	12/22	

Figure 6-10 User Activity Log Details

User	Time	Date	Action Type	Description
OEDS\$OWS	10:26	11/08/18	CHECK IN	Smith-Schuster, Juju has checked in Inspected room 0133 on 11/08/18
OEDS\$OWS	10:26	11/08/18	UPDATE RESERVATION	ROOM -> 0133 RESERVATION TYPE HO -> CHECKED IN
OEDS\$OWS	10:16	11/08/18	UPDATE RESERVATION	WINDOW NUMBER 1 PAYMENT METHOD CA -> MC CREDIT CARD
OEDS\$OWS	10:16	11/08/18	RECORD CREDIT CARD API	APPROVED 184.05 CAD, FOR PAYMENT TYPE MC , APPROVAL CODE
OEDS\$OWS	10:15			
SUPERVISOR	10:11			

User Activity Log Details

User

OEDS\$OWS

Time

10:26

Date

11/08/18

Station ID

UNDEFINED

Action Type

CHECK IN

Smith-Schuster, Juju has checked in Inspected room 0133 on 11/22/17

Report

Details

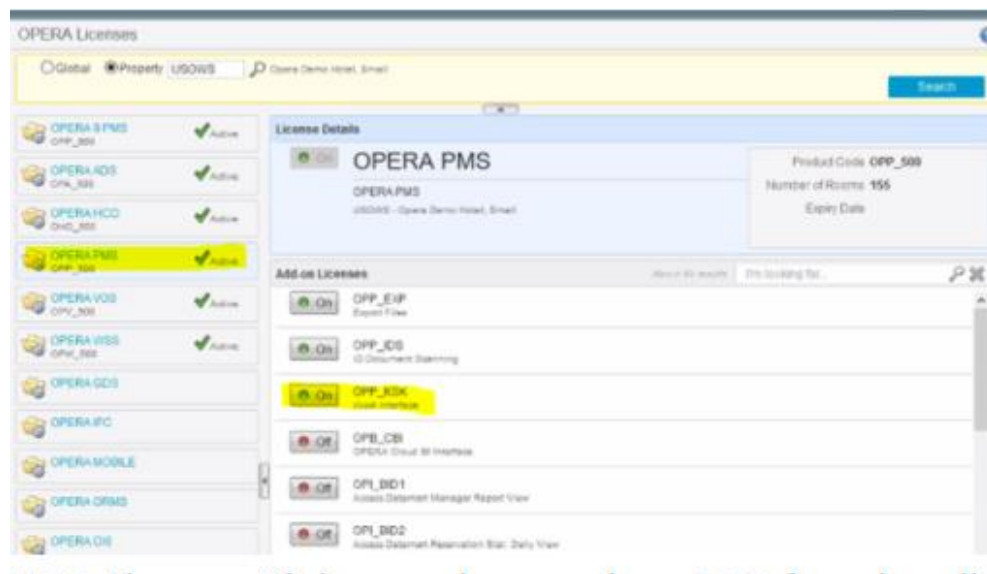
Close

Kiosk Install for 1.20 Environments

1.20 Environment - License Activation

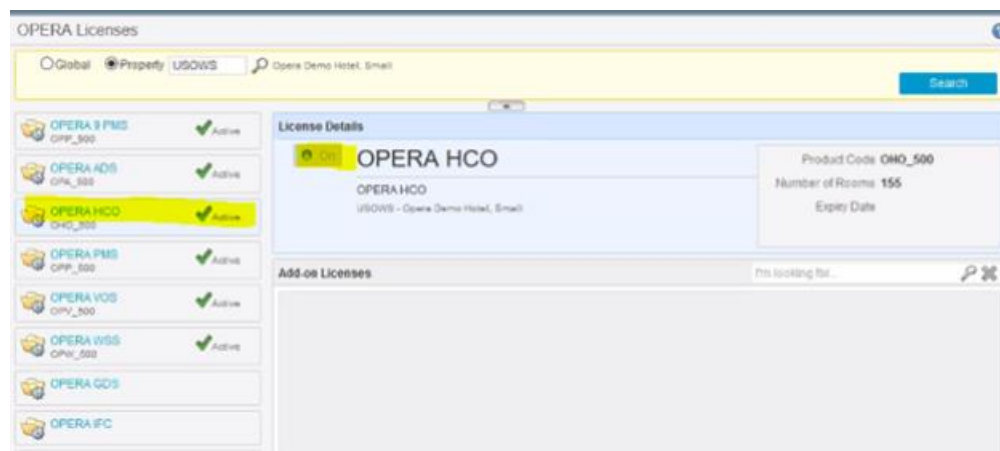
1. Kiosk License: Toolbox > OPERA Licenses > PMS > Activate OPP_KSK:

Figure 7-1 Activate OPP_KSK



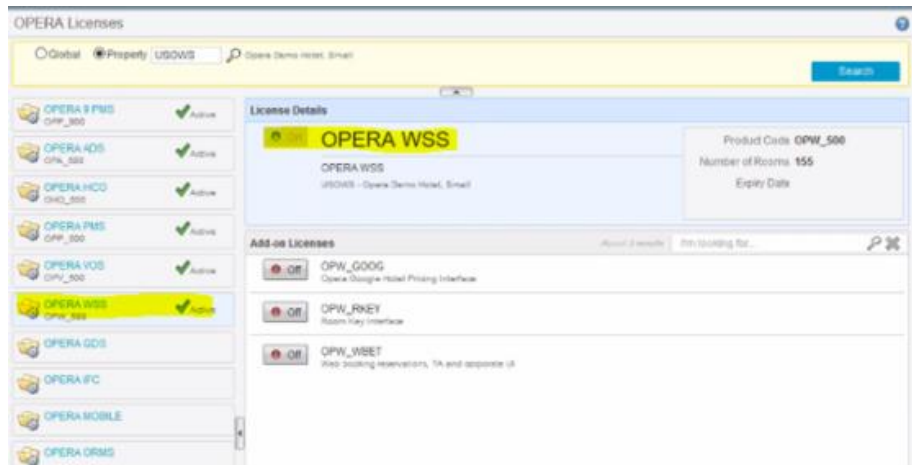
2. HCO License (If the vendor requires O2G functionality for OPI connectivity):
Toolbox > OPERA Licenses > HCO:

Figure 7-2 License Details



3. OWS License: Toolbox > OPERA Licenses > WSS:

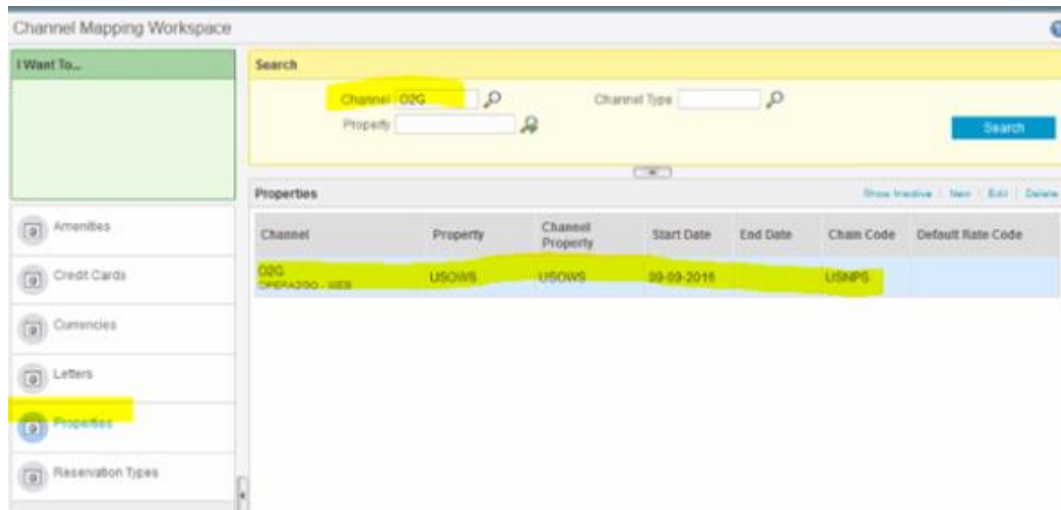
Figure 7-3 License Details



1.20 Environment - Functionality Setup

1. Administration > Interfaces > Channel Mapping > Properties. Verify Mapping exists for O2G Channel:

Figure 7-4 Channel Mapping Workspace



2. Complete the rest of the setup via V5 configuration screens as noted in the manual:
 - a. Setup > Kiosk Configuration > KIOSK Channel
 - b. Setup > Kiosk Configuration > O2G Channel
 - c. Application Settings > OWS group > Settings > Fetch_Reservation parameter to ALL

IFC Setup > OPI Interface > Workstation Setup > Follow setup instructions for the V5 screens.

8

Kiosk Install for OPERA Cloud 20.X Environments

NOTE:

If the Kiosk vendor is invoking the AddPayment function via the O2G channel, then the installation cannot be completed until the release and install of 20.2. If the vendor is not using this functionality, then please proceed with the following steps.

OPERA Cloud 20.X - License Activation



1. **Kiosk License: Toolbox > System Setup > OPERA Licenses > Select property > click SEARCH button >**
2. Select '**OPERA Cloud PMS**' > Scroll down and activate 'OPP_KSK':

Figure 8-1 Kiosk Interface Activation



3. **OWS license: Toolbox > System Setup > OPERA Licenses > Select property > click SEARCH button > Activate 'OPERA Cloud WSS':**

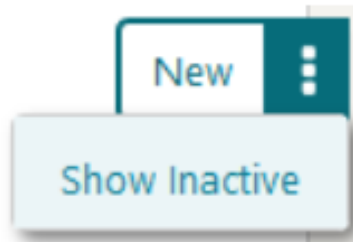
Figure 8-2 OPERA Cloud WSS

Licenses	Active	License Details	
OPERA CLOUD PMS	<input checked="" type="checkbox"/>	 OPERA CLOUD WSS	
OPP_900	<input checked="" type="checkbox"/>		
OPERA CLOUD WSS	<input checked="" type="checkbox"/>	 OPERA CLOUD WSS	
OPW_900	<input checked="" type="checkbox"/>		
OPERA CLOUD ADS	<input type="checkbox"/>	Product Code	
OPERA CLOUD SEM	<input type="checkbox"/>		
OPERA IFC	<input type="checkbox"/>	Number of Rooms	
OPERA ORMS	<input type="checkbox"/>	OPW_900	
OPERA OXI	<input type="checkbox"/>	155	

OPERA Cloud 20.X - Functionality Setup

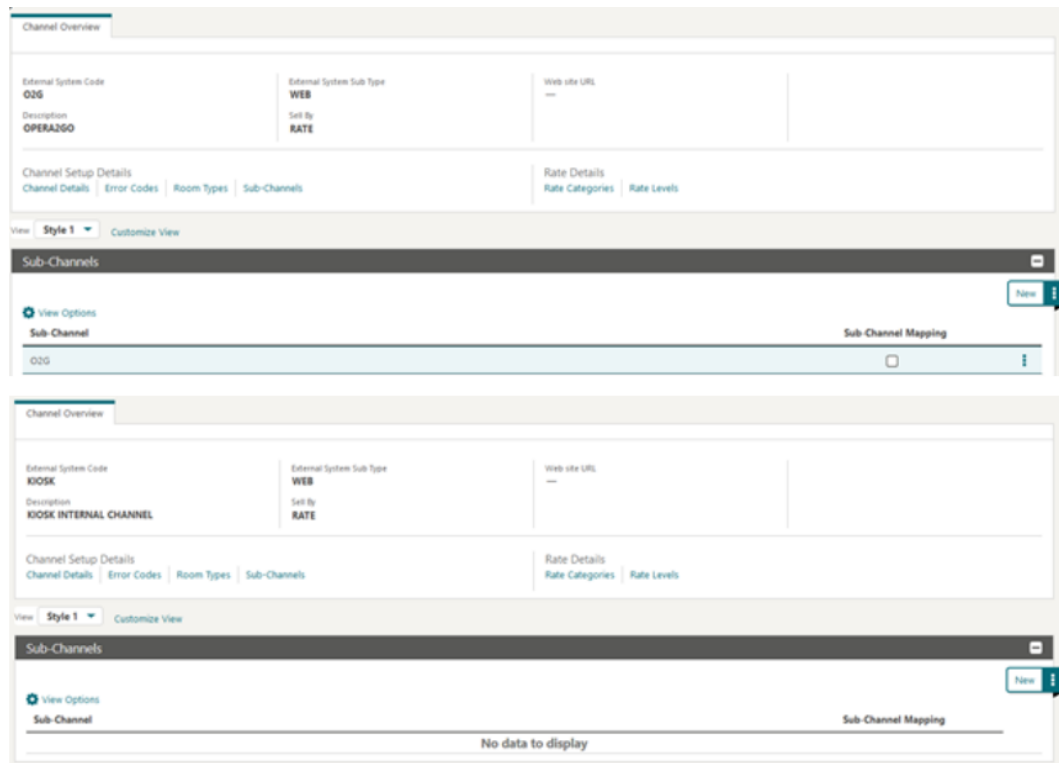
1. **Administration > Interfaces > Channel Configuration > Channel Setup** > Select the 3 dots next to New' and click '**Show Inactive**':

Figure 8-3 Show Inactive



2. Verify if O2G and KIOSK exist and are inactive. If inactive, Edit and add sub-channel mapping of same type (Kiosk does not need sub-channel). If not inactive and does not exist, create as shown below:

Figure 8-4 Channel Overview



3. **Administration > Interfaces > Channel Controls** > Select the KIOSK channel and Kiosk group and configure per the manual.

Figure 8-5 Search Channel

4. **Administration > Interfaces > Channel Mapping > Properties** > Click NEW and add mapping for KIOSK and O2G:

Figure 8-6 Property Management

5. **Administration > Enterprise > OPERA Controls > Channel Management** > change 'Fetch Reservation' to ALL:

Figure 8-7 Fetch Reservation

6. Create a cashier in the front end for the chain KIOSK user. **Administration > Financial > Cashiering Management > Cashiers**:
 - a. One for the chain:

Figure 8-8 Manage Cashier

- b. Note the Cashier ID as this is needed for the script that needs to be run to activate the kiosk_cashier_yn flag.
7. Terminal Setup: **Administration > Interfaces > Property Interfaces > Edit the OPI interface > Workstation Setup >**
 - a. The Terminal maps to the encoder on the PSP side of OPI(if Middleware is used for the OPI vendor, then it needs to be same as the terminal ID as set in middleware). The **Terminal is limited to 16 characters just like the front desk setup.**
 - b. The Device IP/Encoder Number can be same as the Terminal, however due to [BUG 31108396](#), the value must be 10 characters or less. It is advised to use a numerical scheme, starting at 1 and going up on subsequent devices.
 - c. Location: Give a useful name
 - d. Type: must be M

Figure 8-9 Workstation setup

NOTE:

For environments below 20.2 ONLY: The Cloud 20.1 script needs to be run by provisioning to create the cashier, then link cashier to the app_user and add the record to the user_Resort_access table. The process for this is documented at PM OPERA Cloud Implementations.

9

FAQ & Sample XML Messages

OWS messages for Kiosk use

Information.QueryLOV

The Information QueryLov web service allows the list of configured terminals to be returned.

Request	Response
<pre> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" <soap:Header> <OGHeader transactionID="6fa734fa-b5e8-4b31-a6dc-79c0dbc11251" timeStamp="2016-02-12T08:46:49.4886644+01:00" channelValidation="false" xmlns="http://webservices.micros.com/og/4.3/Core/"> <Origin entityID="O2G" systemType="PMS" /> <Destination entityID="O2G" systemType="PMS" /> </OGHeader> </soap:Header> <soap:Body> <LovRequest xmlns="http://webservices.micros.com/ows/5.1/Information.wsdl"> <LovQuery2> <LovIdentifier>CREDIT_CARD_SWIPE_TERMINAL</LovIdentifier> <LovQueryQualifier> </pre>	<pre> <?xml version="1.0" encoding="utf-8"?> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"> <soap:Header> <OGHeader transactionID="6fa734fa-b5e8-4b31-a6dc-79c0dbc11251" timeStamp="2016-02-12T02:46:49.4886644-05:00" primaryLangID="E" channelValidation="false" xmlns="http://webservices.micros.com/og/4.3/Core/"> <Origin entityID="O2G" systemType="PMS"/> <Destination entityID="O2G" systemType="PMS"/> </OGHeader> </soap:Header> <soap:Body> <wsa:Action>http://webservices.micros.com/ows/5.1/Information.wsdl#QueryLovResponse</wsa:Action> <wsa:MessageID>urn:uuid:afb36135-57ee-4b29-af3e-4ebdf9dbfa23</wsa:MessageID> <wsa:RelatesTo>urn:uuid:609c1ebf-c78c-4b03-9d9c-41a8527c8216</wsa:RelatesTo> <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To> </soap:Body> </pre>

Request	Response
<pre>qualifierType="HOST">O2G</LovQueryQualif ier> <LovQueryQualifier qualifierType="RESORT">FSDH</LovQuery Qualifier> </LovQuery2> </LovRequest> </soap:Body> </soap:Envelope</pre>	<pre><LovResponse xmlns:c="http://webservices.micros.com/og /4.3/Core/" xmlns:hc="http://webservices.micros.com/o g/4.3/HotelCommon/" xmlns="http://webservices.micros.com/ows /5.1/Information.wsdl"> <Result resultStatusFlag="SUCCESS"/> <LovQueryResult qualifierType="INTERFACE" qualifierValue="OP01"> <LovValue description="Mobile 1">MOBILE1</LovValue> <LovValue description="Mobile 2">MOBILE2</LovValue> <LovValue description="Mobile Concierge">MOBCONCIERGE</LovValue > <LovValue description="Mobile Valet">MOBVALET</LovValue> </LovQueryResult> </LovResponse> </soap:Body> </soap:Envelope></pre>

ResvAdvanced.AddPayment

The ResvAdvanced AddPayment web service activates the requested OPI vendor device for the guest to dip, swipe or manually enter the credit card. The Credit Card is then authorized and the authorization stored on the reservation. The interface uses the auth rule/amount that is on the reservation or the external system can pass in the desired auth rule/amount to be used.

Request	Response
Request with attached deposit rule on reservation (property default) <pre> <?xml version="1.0" encoding="utf-8"?> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <soap:Header> <OGHeader transactionID="e0bcacb2-cf08-40d4-a13a-5877583d7fc3" timeStamp="2017-01-12T12:30:54.7921314+01:00" primaryLangID="E" channelValidation="false" xmlns="http://webservices.micros.com/og/4.3/Core/"> <Origin entityID="O2G" systemType="PMS"/> <Destination entityID="O2G" systemType="PMS"/> <Authentication> <UserCredentials> <UserName>XXXX</UserName> <UserPassword>XXXX</UserPassword> <Domain>FSDH</Domain> </UserCredentials> </Authentication> </OGHeader> </soap:Header> </soap:Body> </pre>	<pre> <?xml version="1.0" encoding="utf-8"?> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"> <soap:Header> <OGHeader transactionID="6fa734fa-b5e8-4b31-a6dc-79c0dbc11251" timeStamp="2016-02-12T02:46:49.4886644-05:00" primaryLangID="E" channelValidation="false" xmlns="http://webservices.micros.com/og/4.3/Core/"> <Origin entityID="O2G" systemType="PMS"/> <Destination entityID="O2G" systemType="PMS"/> <Authentication> <UserCredentials> <UserName>OEDS\$OWS</UserName> <UserPassword>\$\$\$OEDS\$OWS\$\$</UserPassword> <Domain>FSDH</Domain> </UserCredentials> </Authentication> </OGHeader> <wsa:Action>http://webservices.micros.com/ows/5.1/ResvAdvanced.wsdl#AddPaymentResponse</wsa:Action> <wsa:MessageID>urn:uuid:afb36135-57ee-4b29-af3e-4ebdf9dbfa23</wsa:MessageID> <wsa:RelatesTo>urn:uuid:609c1ebf-c78c-4b03-9d9c-41a8527c8216</wsa:RelatesTo> <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To> </soap:Header> </soap:Body> </pre>

Request	Response
<pre> <AddPaymentRequest xmlns="http://webservices.micros.com/og/4.3/ResvAdvanced/"> <HotelReference chainCode="CHA" hotelCode="FSDH"/> <ResvNameID type="EXTERNAL" source="OPERA_RESV_ID">3388424</ResvNameID> <PaymentType>MC</PaymentType> <TerminalCode>MOBILE1</TerminalCode> <Window>1</Window> <Action>KIOSK CHECKIN</Action> </AddPaymentRequest> </soap:Body> </soap:Envelope> </pre>	<pre> <AddPaymentResponse xmlns="http://webservices.micros.com/og/4.3/ResvAdvanced/"> <Result resultStatusFlag="SUCCESS"/> <PaymentType>MC</PaymentType> <CreditCard cardType="MC"> <cardID xmlns="http://webservices.micros.com/og/4.3/Common/">36910</cardID> <cardNumber xmlns="http://webservices.micros.com/og/4.3/Common/">0009abcdef46ghij</cardNumber> <expirationDate xmlns="http://webservices.micros.com/og/4.3/Common/">2022-12-31</expirationDate> </CreditCard> <ApprovalCode>0A123B </ApprovalCode> <ApprovalAmount>184.05</ApprovalAmount> </AddPaymentResponse> </soap:Body> </soap:Envelope> </pre>
<p>Request with specified deposit rule</p> <pre> <?xml version="1.0" encoding="utf-8"?> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <soap:Header> <OGHeader transactionID="e0bcacb2-cf08-40d4-a13a-5877583d7fc3" timeStamp="2017-01-12T12:30:54.7921314+01:00 </pre>	

Request	Response
<pre> " primaryLangID="E" channelValidation="false" xmlns="http://webservices.micros.com/og /4.3/Core/"> <Origin entityID="O2G" systemType="PMS"/> <Destination entityID="O2G" systemType="PMS"/> <Authentication> <UserCredentials> <UserName>XXXX</UserName> <UserPassword>XXXX</UserPassword> <Domain>FSDH</Domain> </UserCredentials> </Authentication> </OGHeader> </soap:Header> <soap:Body> <AddPaymentRequest xmlns="http://webservices.micros.com/og /4.3/ResvAdvanced/"> <HotelReference chainCode="CHA" hotelCode="FSDH"/> <ResvNameID type="EXTERNAL" source="OPERA_RESV_ID">3388424</ ResvNameID> <PaymentType>AX</PaymentType> <TerminalCode>MOBILE1</TerminalCod e> <Window>1</Window> <res:AuthorizationRule> <!--Optional:--> <res:Rule>5</res:Rule> <!--Optional:--> <!--Pass Amount of Percent based on the authorization rule -- > <res:Amount>60</res:Amount> <!--Optional:--> <!-- <res:Percentage>10</res:Percentage> -- > </res:AuthorizationRule> <res:Action>KIOSK CHECKIN</res:Action> </AddPaymentRequest> </soap:Body> </soap:Envelope> </pre>	

ResvAdvanced.CheckIn

The CheckIn call is executed with a null CreditCard object as the Credit Card has already been authorized.

Request	Response
<pre> <?xml version="1.0" encoding="UTF-8"?> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <soap:Header> <OGHeader transactionID="09384370" timeStamp="2009-02-20T09:34:22.9468750-05:00" xmlns="http://webservices.micos.com/og/4.3/Core/"> <Origin entityID="KIOSK" systemType="KIOSK"/> <Destination entityID="KIOSK" systemType="KIOSK"/> <Authentication> <UserCredentials> <UserName>XXXX</UserName> <UserPassword>XXXX</UserPassword> <Domain>FSDH</Domain> </UserCredentials> </Authentication> </OGHeader> </soap:Header> <soap:Body> <CheckInRequest xmlns="http://webservices.micos.com/og/4.3/ResvAdvanced/" xmlns:c="http://webservices.micos.com/og/4.3/Common/" xmlns:hc="http://webservices.micos.com/og/4.3/HotelCommon/" xmlns:n="http://webservices.micos.com/og/4.3/Name/" xmlns:r="http://webservices.micos.com/og/4.3/Reservation/" </pre>	<pre> <?xml version="1.0" encoding="UTF-8"?> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <soap:Header> <OGHeader transactionID="09384370" timeStamp="2009-02-20T09:34:22.9468750-05:00" xmlns="http://webservices.micos.com/og/4.3/Core/"> <Origin entityID="KIOSK" systemType="KIOSK"/> <Destination entityID="KIOSK" systemType="KIOSK"/> <Authentication> <UserCredentials> <UserName>XXXX</UserName> <UserPassword>XXXX</UserPassword> > <Domain>FSDH</Domain> </UserCredentials> </Authentication> </OGHeader> </soap:Header> <soap:Body> </pre>

Request	Response
<pre> xmlns:ra="http://webservices.micros.com/ og/4.3/ResvAdvanced/" xmlns:xsi="http://www.w3.org/2001/XML Schema-instance"> <ra:ReservationRequest> <ra:HotelReference chainCode="CHA" hotelCode="FSDH"/> <ra:ReservationID> <c:UniqueID type="EXTERNAL" source="RESV_NAME_ID">83400</c:Un iqueID> </ra:ReservationID> </ra:ReservationRequest> <ra:CreditCardInfo/> </CheckInRequest> </soap:Body> </soap:Envelope> </pre>	<pre> <CheckInResponse xmlns:c="http://webservices.micros.com/ og/4.3/Common/" xmlns:hc="http://webservices.micros.co m/og/4.3/HotelCommon/" xmlns="http://webservices.micros.com/o g/4.3/ResvAdvanced/"> <CheckInComplete GuestRegistration="D:\micros\opera\ope raias\webtemp\opera\sample_registratio n_card6311319.pdf"> <ReservationID> <c:UniqueID type="EXTERNAL">83400</c:UniqueID> </ReservationID> <Room> <hc:RoomNumber>7082</hc:RoomNum ber> <hc:RoomDescription> <hc:Text>Pool Side - Queen</hc:Text> </hc:RoomDescription> <hc:RoomType roomTypeCode="POQB"/> </Room> </CheckInComplete> <Result resultStatusFlag="SUCCESS"/> </CheckInResponse> </soap:Body> </soap:Envelope> </pre>

ResvAdvanced.CheckOut

The CheckOut call is executed with a null CreditCard object as the Credit Card has already been authorized.

Request	Response
<pre> <?xml version="1.0" encoding="UTF-8"?> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <soap:Header> <OGHeader transactionID="09384370" timeStamp="2009-02-20T09:34:22.9468750-05:00" xmlns="http://webservices.micros.com/og/4.3/Core/"> <Origin entityID="KIOSK" systemType="KIOSK"/> <Destination entityID="KIOSK" systemType="KIOSK"/> <Authentication> <UserCredentials> <UserName>XXXX</UserName> <UserPassword>XXXX</UserPassword> <Domain>FSDH</Domain> </UserCredentials> </Authentication> </OGHeader> </soap:Header> <soap:Body> <CheckoutRequest xmlns="http://webservices.micros.com/og/4.3/ResvAdvanced/" xmlns:c="http://webservices.micros.com/og/4.3/Common/" xmlns:hc="http://webservices.micros.com/og/4.3/HotelCommon/" xmlns:n="http://webservices.micros.com/og/4.3/Name/" xmlns:r="http://webservices.micros.com/og/4.3/Reservation/" xmlns:ra="http://webservices.micros.com/og/4.3/ResvAdvanced/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <ra:ReservationRequest> <ra:HotelReference chainCode="CHA" hotelCode="FSDH"/> <ra:ReservationID> <c:UniqueID type="EXTERNAL" source="RESV_NAME_ID">83400</c:UniqueID> </ra:ReservationID> </ra:ReservationRequest> </pre>	<pre> <?xml version="1.0" encoding="utf-8"?> <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"> <soap:Header> <OGHeader transactionID="3987870" timeStamp="2009-02-20T09:34:22.946875-05:00" primaryLangID="E" xmlns="http://webservices.micros.com/og/4.3/Core/"> <Origin entityID="KIOSK" systemType="KIOSK"/> <Destination entityID="KIOSK" systemType="KIOSK"/> <Authentication> <UserCredentials> <UserName>XXXX</UserName> <UserPassword>XXXX</UserPassword> <Domain>FSDH</Domain> </UserCredentials> </Authentication> </OGHeader> </soap:Header> <wsa:Action>http://webservices.micros.com/ows/5.1/ResvAdvanced.wsdl#CheckoutResponse</wsa:Action> <wsa:MessageID>urn:uuid:13cdf12a-ae8c-43e0-8d89-01ce9ac22ea2</wsa:MessageID> <wsa:RelatesTo>urn:uuid:cb58abf8-56e5-4f46-b026-f9a6e39ed00c</wsa:RelatesTo> <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To> <wsse:Security> </pre>

Request	Response
<pre> <ra:CreditCardInfo/> </CheckOutRequest> </soap:Body> </soap:Envelope> </pre>	<pre> <wsu:Timestamp wsu:Id="Timestamp- 51d1ba4b-571a-4c43-a2ae- 92ada64467b3"> <wsu:Created>2009-03- 31T14:03:28Z</wsu:Created> <wsu:Expires>2009-03- 31T14:08:28Z</wsu:Expires> </wsu:Timestamp> </wsse:Security> </soap:Header> <soap:Body> <CheckOutResponse xmlns:c="http://webservices.micros.com/ og/4.3/Common/" xmlns:hc="http://webservices.micros.co m/og/4.3/HotelCommon/" xmlns="http://webservices.micros.com/o g/4.3/ResvAdvanced/"> <CheckOutComplete> <ReservationID> <c:UniqueID type="INTERNAL" source="OPERA_RESV_ID">1318912</ c:UniqueID> </ReservationID> </CheckOutComplete> <Result resultStatusFlag="SUCCESS"/> </CheckOutResponse> </soap:Body> </soap:Envelope> </pre>