

Implement CPQ-EBS Inventory On Hand Balance Integration

ORACLE TECHNICAL WHITE PAPER | CPQ 2015 R1 RELEASE | 2015



Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



Table of Content

Purpose	1
Prerequisite.....	1
Introduction	1
Implementation Overview	1
FOM Web Services	1
Commerce Process.....	1
Additional Changes	2
Implementation Details	2
Attributes	2
File Manager	2
Data Table.....	2
Library Functions.....	3
Actions.....	8
Step Transitions	8
Appendix A: Inventory On Hand Balance Template	9
Appendix B: Query Quantity BML.....	12
Appendix C: Sample Query Quantity Payload	17

Purpose

This white paper describes how to implement the Reference Integration between Oracle Configure, Price and Quote (CPQ) Cloud and Oracle E-Business Suite (EBS) Inventory On-Hand Balance.

Prerequisite

The CPQ – EBS Reference Integration is implemented on CPQ Cloud Release 2015 R1 or later of Oracle E-Business Suite Order Inventory On-Hand Balance. CPQ base reference application must be deployed in the environment. Relevant E-Business Suite Inventory On-Hand Balance web services must be up and running.

Introduction

CPQ – EBS Reference Integration enables CPQ to check the availability of an item. Even though there is no validation preventing the creation of an order, it can be added per the customer's needs.

Implementation Overview

FOM Web Services

For FOM order creation, the web service invoked is "INV_QUANTITY_TREE_PUB_Port" operation "UPDATE_QUANTITIES".

Web Service Generation:

1. Login to the EBS environment.
2. Navigate to the web service as follows:
 - a) **UPDATE_QUANTITIES**: Integrated SOA gateway > Integration Repository > Supply Chain Management > Inventory Management > Inventory On Hand Balance > Quantity tree Application Program Interface
3. Check the "UPDATE_QUANTITIES" operation under procedures and functions to generate the corresponding web service.
4. Click **Generate**.
A WSDL link for the SOAP web service is generated.
5. Select **Username Token** as the **Authentication Type**.
6. Click **Deploy**.
7. Click View WSDL and copy the value of the "soap:address location" element, which is the endpoint.
8. Select the **Grants** tab.
9. Select the object for which the grant is created.
10. Keep the **Grantee Type** as **All Users**.
11. Click **Create Grant**.

Commerce Process

The Reference Integration makes changes to the "Oracle Quote to Order" commerce process from the base reference application in the following areas:

- » **Attributes**
- » **Actions**
- » **Library Functions**
- » **Step Transitions**

Additional Changes

Additional changes outside the Commerce process include:

- » **Data Table**
- » **File Manager**

Implementation Details

Attributes

Create the following attributes for the “Oracle Quote to Order” Commerce process.

Document	Attribute Name	Attribute Variable Name	Type	Description
Transaction Line	Availability Status	availabilityStatus	Text	Sets status as “Available”, “Not Available” or “Not Checked”
Transaction Line	Available Units	availableUnits	Text	Updates the available units for the items
Transaction Header	Integration Status	integrationStatusTransaction	Text	Shows the status of the integration process
Transaction Header	Warehouse Organisation ID	warehouseOrgID	Text	Used to set value of the warehouse organization ID

File Manager

Template file format information is located in the below table.

Folder	Description	Template Files
EBS	Inventory on Hand Balance	Refer Appendix A: Inventory On Hand Balance Template

Data Table

INT_SYSTEM_DETAILS

This table contains the information regarding the EBS-Order Management end point.

System(Ke)	UserName	WebService	SoapEndPoint	RestEndPoint
QUERY_ON_HAND	<Enter the username here to call the webservice endpoint>	<Enter the webservice SOAP/REST>	<Enter the SOAP webservice endpoint to call the service related to EBS-OM>	<Enter the REST webservice endpoint to call the service related to EBS-OM>

INT_SYSTEM_TEMPLATES

This table contains the information regarding the supporting template stored in File Manager.

System(Key)	Operation(Key)	Template
EBS	queryOnHandSoap	<Enter the template URL path that is uploaded in File manager>

Library Functions

The Commerce process library functions used are identified in this section:

getPassword

This library function provides the password to get templates.

```
pswd = "";

if ( sysName == "EBS" ) {
    pswd = "welcome";
}

if ( sysName == "MATERIAL RESERVATION" ) {
    pswd = "sysadmin";
}

if ( sysName == "QUERY_ON_HAND" ) {
    pswd = "sysadmin";
}

if ( sysName == "CLEAR_CACHE" ) {
    pswd = "sysadmin";
}

if ( sysName == "EBS-Customer" ) {
    pswd = "sysadmin";
}

if ( sysName == "EBS-Address" ) {
    pswd = "sysadmin";
}

return pswd;
```

Return Type and Input information for the library function are shown below.

Commerce BML Library Function Editor: Properties & Parameters

Name:	getPassword	#	Parameter Name	Parameter Type
Variable Name:	getPassword	1	sysName	String
Description:				
Return Type:	String			

Function Editor

Hide Tools | Editor Help

Attributes | Function Wizard | Debugger | Library Function(s)

Main Document Attribute	Sub Document Attribute	System Attribute
# Attribute	transactionLine	# Attribute
# Attribute	# Attribute	

getTemplateLocation

This library function gets the template location.

```
//1. Get Template File
templateUrl = "";
//bmq1 query
resultSet = bmq1("Select Template from INT_SYSTEM_TEMPLATES where System = $system
and Operation = $operation");

//loop through the records
for record in resultSet {
    templateUrl = get(record,"Template");
    print templateUrl;
}

temp=split(templateUrl,"image");

return temp[1];
```

Return Type and Input information for the library function are shown below.

Commerce BML Library Function Editor: Properties & Parameters

Name:	getTemplateLocation	#	Parameter Name		Parameter Type
Variable Name:	getTemplateLocation	1	system	↓	String
Description:		2	operation	↓	String
Return Type:	String				

Function Editor

Hide Tools | 🔍 🗨️ Editor Help

🔗 Attributes | 🔧 Function Wizard | 🐛 Debugger | 📁 Library Function(s)

Main Document Attribute	Sub Document Attribute	System Attribute
# Attribute	transactionLine ↓	# Attribute
	# Attribute	

getDataSecurity

This library function is used for getting the data security parameters for the SOAP header request.

```
headerFile_mod=headerFile;
// For Material Reservation
if(system == "MATERIAL RESERVATION" AND operation == "Reserve"){
    headerFile_mod= replace(headerFile_mod , "{{Responsibility_Name}}", "System
Administrator");
    headerFile_mod= replace(headerFile_mod , "{{Application_Name}}", "SYSADMIN");
    headerFile_mod= replace(headerFile_mod , "{{Security_Group_Name}}", "STANDARD");
    headerFile_mod= replace(headerFile_mod , "{{Language}}", "AMERICAN");
    headerFile_mod= replace(headerFile_mod , "{{Org_Id}}", "204");
}
if(system == "MATERIAL RESERVATION" AND operation == "Relieve"){
    headerFile_mod= replace(headerFile_mod , "{{Responsibility_Name}}", "System
Administrator");
    headerFile_mod= replace(headerFile_mod , "{{Application_Name}}", "SYSADMIN");
    headerFile_mod= replace(headerFile_mod , "{{Security_Group_Name}}", "STANDARD");
    headerFile_mod= replace(headerFile_mod , "{{Language}}", "AMERICAN");
    headerFile_mod= replace(headerFile_mod , "{{Org_Id}}", "204");
}
if(system == "EBS" AND operation == "queryOnHandSoap"){
    headerFile_mod= replace(headerFile_mod , "{{Responsibility_Name}}", "System
Administrator");
    headerFile_mod= replace(headerFile_mod , "{{Application_Name}}", "SYSADMIN");
    headerFile_mod= replace(headerFile_mod , "{{Security_Group_Name}}", "STANDARD");
    headerFile_mod= replace(headerFile_mod , "{{Language}}", "AMERICAN");
    headerFile_mod= replace(headerFile_mod , "{{Org_Id}}", "204");
}
return headerFile_mod;
```

Return Type and Input information for the library function are shown below.

Commerce BML Library Function Editor: Properties & Parameters

Name:	<input type="text" value="getDataSecurity"/>		#	Parameter Name	Parameter Type
Variable Name:	<input type="text" value="getDataSecurity"/>		1	system	↓ String
Description:	<input type="text"/>		2	operation	↓ String
			3	headerFile	↓ String
Return Type:	<input type="text" value="String"/>				

Function Editor

Hide Tools |
Editor Help

Attributes
 Function Wizard
 Debugger
 Library Function(s)

Main Document Attribute	Sub Document Attribute	System Attribute
# Attribute	transactionLine ↓ # Attribute	# Attribute

Actions

Query Quantity

The integration is implemented using a single action, "Query Quantity", as shown below.

System Variable Name	Type	Description
N/A		
Variable Name for (Transaction)	Type	Description
bs_id	String	Transaction ID
orgID	String	Organization ID
warehouseOrgID	String	Warehouse organization ID
Variable Name for (Transaction Line)	Type	Description
transactionLine	Collection of Sub Documents	
_document_number	String	Document Number
_part_number	String	Part Number
lineNumber_I	String	Line #
requestedQuantity_I	Integer	Quantity
quantityStatus	String	Availability Status
Imported Commerce Functions		
String getDataSecurity(String system, String operation, String headerFile)		
String getTemplateLocation(String system, String operation)		
String getPassword(String sysName)		

IMPORTANT: Refer to [Appendix B: Query Quantity BML](#) for the Query Quantity action. Define the action using Advanced Modify – After Formula section.

Step Transitions

Action Variable Name	Available Step	Comments
queryQuantity (Action : Query Quantity)	Displayed in all steps from "Approved"	Query Quantity is displayed only when the order is approved.

Sample Payload

Refer to [Appendix C: Sample Query Quantity Payload](#).

Appendix A: Inventory On Hand Balance Template

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:inv="http://xmlns.oracle.com/apps/inv/soapprovider/plsql/inv_quantity_tree_pub/"
"
xmlns:quer="http://xmlns.oracle.com/apps/inv/soapprovider/plsql/inv_quantity_tree_pub
/query_quantities/">
```

```
<soapenv:Header>
```

```
<inv:SOAHeader>
```

```
<inv:RESPONSIBILITY_NAME>{{RESPONSIBILITY_NAME}}</inv:RESPONSIBILITY_NAME>
```

```
<inv:RESPONSIBILITY_APPL_NAME>{{RESPONSIBILITY_APPL_NAME}}</inv:RESPONSIBILITY
_APPL_NAME>
```

```
<inv:SECURITY_GROUP_NAME>{{SECURITY_GROUP_NAME}}</inv:SECURITY_GROUP_NAME>
```

```
<inv:NLS_LANGUAGE>{{NLS_LANGUAGE}}</inv:NLS_LANGUAGE>
```

```
<inv:ORG_ID>{{ORG_ID}}</inv:ORG_ID>
```

```
</inv:SOAHeader>
```

```
<wsse:Security soapenv:mustUnderstand="1"
```

```
xmlns:env="http://schemas.xmlsoap.org/soapenv/envelope/" xmlns="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
```

```
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd"><wsse:UsernameToken xmlns="http://docs.oasis-
```

```
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
```

```
<wsse:Username>{{USERNAME}}</wsse:Username>
```

```
<wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">{{PASSWORD}}</wsse:Password>
```

```
</wsse:UsernameToken>
```

```
</wsse:Security>
```

```
</soapenv:Header>
```

```
<soapenv:Body>
```

```
<quer:InputParameters>
```

```
<quer:P_API_VERSION_NUMBER>{{P_API_VERSION_NUMBER}}</quer:P_API_VERSION_NUMBER
```

```
>
```

```
<quer:P_INIT_MSG_LST>{{P_INIT_MSG_LST}}</quer:P_INIT_MSG_LST>
```

<quer:P_ORGANIZATION_ID>{{P_ORGANIZATION_ID}}</quer:P_ORGANIZATION_ID>

<quer:P_INVENTORY_ITEM_ID>{{P_INVENTORY_ITEM_ID}}</quer:P_INVENTORY_ITEM_ID>

<quer:P_TREE_MODE>{{P_TREE_MODE}}</quer:P_TREE_MODE>

<quer:P_IS_REVISION_CONTROL></quer:P_IS_REVISION_CONTROL>

<quer:P_IS_LOT_CONTROL></quer:P_IS_LOT_CONTROL>

<quer:P_IS_SERIAL_CONTROL></quer:P_IS_SERIAL_CONTROL>

<quer:P_DEMAND_SOURCE_TYPE_ID>{{P_DEMAND_SOURCE_TYPE_ID}}</quer:P_DEMAND_SOURCE_TYPE_ID>

<quer:P_DEMAND_SOURCE_HEADER_ID>{{P_DEMAND_SOURCE_HEADER_ID}}</quer:P_DEMAND_SOURCE_HEADER_ID>

<quer:P_DEMAND_SOURCE_LINE_ID>{{P_DEMAND_SOURCE_LINE_ID}}</quer:P_DEMAND_SOURCE_LINE_ID>

<quer:P_DEMAND_SOURCE_NAME></quer:P_DEMAND_SOURCE_NAME>

<quer:P_LOT_EXPIRATION_DATE></quer:P_LOT_EXPIRATION_DATE>


<quer:P_REVISION></quer:P_REVISION>

<quer:P_LOT_NUMBER></quer:P_LOT_NUMBER>

<quer:P_SUBINVENTORY_CODE></quer:P_SUBINVENTORY_CODE>

<quer:P_LOCATOR_ID></quer:P_LOCATOR_ID>

<quer:P_ONHAND_SOURCE>{{P_ONHAND_SOURCE}}</quer:P_ONHAND_SOURCE>



```
<quer:P_TRANSFER_SUBINVENTORY_CODE></quer:P_TRANSFER_SUBINVENTORY_CODE>

    <quer:P_COST_GROUP_ID></quer:P_COST_GROUP_ID>

    <quer:P_LPN_ID></quer:P_LPN_ID>

    <quer:P_TRANSFER_LOCATOR_ID></quer:P_TRANSFER_LOCATOR_ID>
  </quer:InputParameters>
</soapenv:Body>
</soapenv:Envelope>
```

Appendix B: Query Quantity BML

```
system = "EBS";
operation="queryOnHandSoap";

//Header Values
headerValues = dict("string");
put(headerValues, "Content-Type", "text/xml; charset=utf-8");

//Get Endpoint for SOAP request
endpointResultSet = bmql("Select SoapEndPoint from INT_SYSTEM_DETAILS where System =
'QUERY_ON_HAND'");
endpoint = "";
for record in endpointResultSet {
    endpoint = get(record,"SoapEndPoint");
}
print endpoint;
//Payload dictionary
payload = dict("string");
//Get User Info from data table and put in dictionary
resultSet = bmql("Select UserName from INT_SYSTEM_DETAILS where System
='QUERY_ON_HAND'");
for record in resultSet {
    username = get(record,"UserName");
    password = commerce.getPassword("QUERY_ON_HAND");
    put(payload,"USERNAME", username);
    put(payload,"PASSWORD", password);
}
put(payload ,"P_API_VERSION_NUMBER", "1");
put(payload ,"P_INIT_MSG_LST", "T");
put(payload ,"P_ORGANIZATION_ID", warehouseOrgID);
put(payload ,"P_TREE_MODE", "3");
put(payload ,"P_DEMAND_SOURCE_TYPE_ID", "2");
if (headerID <> ""){
```

```

    put(payload , "P_DEMAND_SOURCE_HEADER_ID", headerID);}
else{
    put(payload , "P_DEMAND_SOURCE_HEADER_ID", bs_id);}
put(payload , "P_ONHAND_SOURCE", "1");
put(payload , "P_IS_REVISION_CONTROL", "");
put(payload , "P_IS_LOT_CONTROL", "");
put(payload , "P_IS_SERIAL_CONTROL", "");

result = "";
defaultErrorMessage="";
templateLocation = commerce.getTemplateLocation(system, operation);

//part-numbers of all lines
partNumbers = dict("string");
checkQuantityResponse = "";
quantityCheckPayload = "";

for line in transactionLine{
    //line-specific values added to dictionary
    put(payload, "P_INVENTORY_ITEM_ID", line._part_number);
    put(payload, "P_DEMAND_SOURCE_LINE_ID", line.lineNumber_1);

    //apply template
    quantityCheckPayload
=applytemplate(templateLocation, payload, defaultErrorMessage);

    //add data security to the payload
    quantityCheckPayload = commerce.getDataSecurity(system, operation,
quantityCheckPayload );

    //check quantity
    reservableQuantity = "";
    checkQuantityResponse = urldatabypost(endpoint, quantityCheckPayload, "",
headerValues, true);
    print checkQuantityResponse + "\n\n";

```



```

foundError = find(checkQuantityResponse,"ERROR:");
if( foundError > -1){
    return "1~integrationStatusTransaction~" + checkQuantityResponse +
"|1~soapResponse~" + checkQuantityResponse + "|1~soapRequest~"+quantityCheckPayload
+"|";
}

foundAuthError = find(checkQuantityResponse,"<faultstring>");
if( foundAuthError > -1){
    foundAuthEnd= find(checkQuantityResponse, "</faultstring>");
    authError = substring(checkQuantityResponse, foundAuthError + 13,
foundAuthEnd);

    return "1~integrationStatusTransaction~" + authError +
"|1~soapResponse~" + checkQuantityResponse + "|1~soapRequest~"+quantityCheckPayload
+"|";
}

foundReturnStatus = find(checkQuantityResponse,"<X_RETURN_STATUS>");
if( foundAuthError > -1){
    foundReturnStatusEnd= find(checkQuantityResponse, "</X_RETURN_STATUS>");
    returnStatus = substring(checkQuantityResponse, foundReturnStatus + 17,
foundReturnStatusEnd);

    if(trim(returnStatus) == "E"){
        foundError = find(checkQuantityResponse,"<X_MSG_DATA>");
        foundErrorEnd= find(checkQuantityResponse, "</X_MSG_DATA>");
        error = substring(checkQuantityResponse, foundError + 12,
foundErrorEnd);

        return "1~integrationStatusTransaction~" + error +
"|1~soapResponse~" + checkQuantityResponse + "|1~soapRequest~"+quantityCheckPayload
+"|";
    }
}

foundStartPattern= find(checkQuantityResponse,"<X_ATR>");
if (foundStartPattern > -1){
    foundEndPattern= find(checkQuantityResponse, "</X_ATR>");


```

```

        reservableQuantity = substring(checkQuantityResponse,
foundStartPattern+7, foundEndPattern);
    }
    quantity = line.requestedQuantity_1;
    isExisting = get(partNumbers, line._part_number);
    print isExisting;

    //In BM we might create two lines with the same part-number. In such a case,
the web-service
    //call for each line is independent, even though we need to consider the sum
of the quantities.
    //So a part-numbers dictionary is maintained.
    //if line is not in the partnumbers dictionary, then add it
    if(trim(isExisting) == ""){
        put(partNumbers, line._part_number, string(quantity));
    }
    //if partnumber of line present in dictionary, update the quantity
    else{
        quantity = quantity + atoi(isExisting);
        put(partNumbers, line._part_number, string(quantity));
    }
    isAvailable = "Not Available";
    if(reservableQuantity == ""){
        return "1~integrationStatusTransaction~Error in quantity
retrieval|1~soapResponse~" + checkQuantityResponse +
"|1~soapRequest~"+quantityCheckPayload +"|";
    }
    if(atoi(reservableQuantity) >= quantity){
        isAvailable = "Available";
    }
    result = result + line._document_number + "~quantityStatus~" + isAvailable +
"|";
        + line._document_number + "~availableUnits~" + reservableQuantity +
"|";
}

```




```
result = result + "|1~soapResponse~" + checkQuantityResponse +  
"|1~soapRequest~"+quantityCheckPayload +"|";  
return result;
```


Appendix C: Sample Query Quantity Payload

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:inv="http://xmlns.oracle.com/apps/inv/soapprovider/plsql/inv_quantity_tree_pub/"
"
xmlns:quer="http://xmlns.oracle.com/apps/inv/soapprovider/plsql/inv_quantity_tree_pub
/query_quantities/">
  <soapenv:Header>
    <inv:SOAHeader>
      <inv:RESPONSIBILITY_NAME>System Administrator</inv:RESPONSIBILITY_NAME>
<inv:RESPONSIBILITY_APPL_NAME>SYSADMIN</inv:RESPONSIBILITY_APPL_NAME>
<inv:SECURITY_GROUP_NAME>STANDARD</inv:SECURITY_GROUP_NAME>
<inv:NLS_LANGUAGE>AMERICAN</inv:NLS_LANGUAGE>
      <inv:ORG_ID>204</inv:ORG_ID>
    </inv:SOAHeader>
    <wsse:Security soapenv:mustUnderstand="1"
xmlns:env="http://schemas.xmlsoap.org/soapenv/envelope/" xmlns="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd"><wsse:UsernameToken xmlns="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
      <wsse:Username>sysadmin</wsse:Username>
      <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">sysadmin</wsse:Password>
    </wsse:UsernameToken>
  </wsse:Security>
</soapenv:Header>
<soapenv:Body>
  <quer:InputParameters>
    <!--Optional:-->
<quer:P_API_VERSION_NUMBER>1</quer:P_API_VERSION_NUMBER>
    <!--Optional:-->
    <quer:P_INIT_MSG_LST>T</quer:P_INIT_MSG_LST>
    <!--Optional:-->
<quer:P_ORGANIZATION_ID>207</quer:P_ORGANIZATION_ID>
    <!--Optional:-->
```

```
<quer:P_INVENTORY_ITEM_ID>149</quer:P_INVENTORY_ITEM_ID>
  <!--Optional:-->
  <quer:P_TREE_MODE>2</quer:P_TREE_MODE>
  <!--Optional:-->
<quer:P_IS_REVISION_CONTROL></quer:P_IS_REVISION_CONTROL>
  <!--Optional:-->
  <quer:P_IS_LOT_CONTROL></quer:P_IS_LOT_CONTROL>
  <!--Optional:-->
<quer:P_IS_SERIAL_CONTROL></quer:P_IS_SERIAL_CONTROL>
  <!--Optional:-->
<quer:P_DEMAND_SOURCE_TYPE_ID>2</quer:P_DEMAND_SOURCE_TYPE_ID>
  <!--Optional:-->
<quer:P_DEMAND_SOURCE_HEADER_ID>1000</quer:P_DEMAND_SOURCE_HEADER_ID>
  <!--Optional:-->
<quer:P_DEMAND_SOURCE_LINE_ID>1002</quer:P_DEMAND_SOURCE_LINE_ID>
  <!--Optional:-->
<quer:P_DEMAND_SOURCE_NAME></quer:P_DEMAND_SOURCE_NAME>
  <!--Optional:-->
<quer:P_LOT_EXPIRATION_DATE></quer:P_LOT_EXPIRATION_DATE>
  <!--Optional:-->
  <quer:P_REVISION></quer:P_REVISION>
  <!--Optional:-->
  <quer:P_LOT_NUMBER></quer:P_LOT_NUMBER>
  <!--Optional:-->
<quer:P_SUBINVENTORY_CODE></quer:P_SUBINVENTORY_CODE>
  <!--Optional:-->
  <quer:P_LOCATOR_ID></quer:P_LOCATOR_ID>
  <!--Optional:-->
  <quer:P_ONHAND_SOURCE>3</quer:P_ONHAND_SOURCE>
  <!--Optional:-->
```



```
<quer:P_TRANSFER_SUBINVENTORY_CODE></quer:P_TRANSFER_SUBINVENTORY_CODE>
  <!--Optional:-->
  <quer:P_COST_GROUP_ID></quer:P_COST_GROUP_ID>
  <!--Optional:-->
  <quer:P_LPN_ID></quer:P_LPN_ID>
  <!--Optional:-->
<quer:P_TRANSFER_LOCATOR_ID></quer:P_TRANSFER_LOCATOR_ID>
  </quer:InputParameters>
</soapenv:Body>
</soapenv:Envelope>
```



Copyright © 2018 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 is software or related documentation that 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.

Integrated Cloud Applications & Platform Services