

ORACLE®

CPQ Cloud

# Implement CPQ-EBS Order Management Integration

ORACLE TECHNICAL WHITE PAPER | CPQ 2015 R1 RELEASE | 2015

ORACLE®



## 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 Contents

Disclaimer .....	1
Purpose .....	1
Prerequisite.....	1
Introduction .....	1
Implementation Overview .....	1
EBS-OM Product Items Synchronization .....	1
EBS-OM Web Services .....	1
Commerce Process.....	2
Additional Changes .....	2
Implementation Details .....	3
EBS-OM Product Items Synchronization .....	3
Attributes .....	5
File Manager .....	5
Data Table.....	6
Library Functions.....	7
Actions.....	22
Step Transitions .....	24
Formulas .....	24
Rules .....	25
Sample Payloads .....	26
Appendix A: Create/Update Order Header Template.....	27
Appendix B: Create/Update Order Line Template.....	29
Appendix C: Cancel Order/Cancel Line Template .....	30
Appendix D: BML for Create Order Action .....	32
Appendix E: BML for the Update Order Action.....	33
Appendix F: BML for the Cancel Order Action .....	34
Appendix G: Cancel Line Order Action.....	36
Appendix H: Sample Create Order Payload.....	37
Appendix I: Sample Update Order Payload .....	39
Appendix J: Cancel Order Payload .....	41
Appendix K: Cancel Line Payload .....	43

## Purpose

This white paper describes steps to implement the Reference Integration between Oracle Configure, Price and Quote (CPQ) Cloud and Oracle E-Business Suite (EBS) Order Management.

## Prerequisite

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

## Introduction

CPQ – EBS Reference Integration enables CPQ to send across the details for placing an order in E-Business Suite Order Management. When a sales representative adds relevant products and clicks the action button “Create Order”, the information is sent across to EBS-OM which creates the order based on the details furnished by CPQ. “Update Order” action button updates the order. Order can be updated if some line attribute has been changed or if new lines are added in the order. “Cancel Order” action button cancels the order. “Cancel Line” action button cancels any line in the order.

## Implementation Overview

### EBS-OM Product Items Synchronization

Parts in CPQ are uploaded from the following EBS tables - “qp\_list\_lines”, “qp\_pricing\_attributes”, “mtl\_system\_items”.

### EBS-OM Web Services

For EBS-OM order creation, web service invoked is:

- » Web service “**OE\_INBOUND\_INT\_Service**” operation “processOrder”

EBS Order Management Web Service Generation:

1. Login to the EBS environment.
2. Navigate to the web service as follows:  
**OE\_INBOUND\_INT\_Service**: Integrated SOA gateway > Integration Repository > Order Management Suite > Order Management > Sales Order > OE\_INBOUND\_INT
3. Check the “processOrder” 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**
- » **Formulas**
- » **Rules**

## Additional Changes

Additional changes outside the commerce process include:

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

## Implementation Details

### EBS-OM Product Items Synchronization

- (1) Query all the item details from the EBS environment using the SQL for the given inventory org id. In this example, org id 204 is used.

```
(Select 'add' as "update", products.* from (SELECT
DISTINCT qpa.PRODUCT_ATTR_VALUE "part_number",
mtl.description "description: en",
'One Time' as "_part_custom_field8",
ql.operand "price: USD"
FROM qp_list_lines ql,
qp_pricing_attributes qpa,
mtl_system_items mtl
WHERE ql.list_line_id = qpa.list_line_id
AND qpa.PRODUCT_ATTR_VALUE = mtl.inventory_item_id
AND qpa.PRODUCT_ATTRIBUTE_CONTEXT = 'ITEM'
AND qpa.PRODUCT_ATTRIBUTE = 'PRICING_ATTRIBUTE1'
AND qpa.PRODUCT_UOM_CODE = 'Ea'
AND qpa.list_header_id = '1000'
AND ql.operand >0) products)
UNION(
Select 'update' as "update", update_products.* from (SELECT
DISTINCT qpa.PRODUCT_ATTR_VALUE "part_number",
mtl.description "description: en",
'One Time' as "_part_custom_field8",
ql.operand "price: USD"
FROM qp_list_lines ql,
qp_pricing_attributes qpa,
mtl_system_items mtl
WHERE ql.list_line_id = qpa.list_line_id
AND qpa.PRODUCT_ATTR_VALUE = mtl.inventory_item_id
AND qpa.PRODUCT_ATTRIBUTE_CONTEXT = 'ITEM'
AND qpa.PRODUCT_ATTRIBUTE = 'PRICING_ATTRIBUTE1'
AND qpa.PRODUCT_UOM_CODE = 'Ea'
AND qpa.list_header_id = '1000'
AND ql.operand >0) update_products);
```

(2) Export the query output in a CSV file.

(3) Open the CSV file and enter the header as **Parts**. The file becomes as shown in the below example.

```
Parts,,,,
,,,,
update,part_number,description: en,_part_custom_field8,price: USD
add,55555555,chk,One Time,56
update,9999999,chk,One Time,57
add,10024,Weekly Vision Subscription,One Time,3
update,10024,Weekly Vision Subscription,One Time,3
add,10026,Monthly Vision Subscription,One Time,10
update,10026,Monthly Vision Subscription,One Time,10
add,10028,Annual Vision Subscription,One Time,100
update,10028,Annual Vision Subscription,One Time,100
add,102612,Lancet 200 Pack,One Time,51.8
update,102612,Lancet 200 Pack,One Time,51.8
add,10285,Vision PRO 128 MB Graphics Card,One Time,119.99
update,10285,Vision PRO 128 MB Graphics Card,One Time,119.99
add,10291,Vision PRO Graphics Update,One Time,199.99
update,10291,Vision PRO Graphics Update,One Time,199.99
add,10492,Vision WireFreedom Wireless Modem,One Time,200
.....

//Note: ... The above is list of the item details from the EBS environment using the SQL for
the given inventory org id.(org id 204 is used.) This has only few lines all other items will
be listed like that.
```

(4) Create a zip file with the CSV file.

(5) Upload the CSV file to the BM environment using the following steps:

(i) Under the BM administration Platform, select "**Parts**" from the "**Products**" category.

(ii) Click on the "**Import**" button which is beside the "**Add New Part**" button.

(iii) Browse for the zip file and click on "**Add**".

(iv) Select the zip file shown and click "**Upload**".

(6) Verify the uploaded parts by creating a new transaction with some of the uploaded parts.

## Attributes

We need to create/modify following attributes in the “Oracle Quote to Order” commerce process. The table below shows all the attributes used.

Document	Attribute Name	Attribute Variable Name	Type	Description
Transaction Header	OrderID	headerID	Text	EBS assigns headerID to each order created.
Transaction Header	Order Number	orderNumber	Text	EBS assigns Order Number to each order created.
Transaction Header	Org ID	orgID	Text	Organization ID.
Transaction Header	Integration Status	integrationStatusTransaction	Read-only text or HTML	Displays success or error message after Web Service call. Default value: Data : \$integrationStatusTransaction\$
Transaction Line > Item Attributes	Part Number	_part_number	Text	Part Number
Transaction Line	priceList	priceList	Text	The list price of the item.
Transaction Line	Cancelled Quantity	cancelledQuantity	Integer	The quantity of any item cancelled by customer.
Transaction Line	lineID	lineId	Text	Transaction Line Id
Transaction Line	_document_number	_document_number	Integer	Document Number
Transaction Header	Soap Request	soapRequest	Text Area	SOAP Request (for debugging purpose)
Transaction Header	Soap Response	soapResponse	Text Area	SOAP Response (for debugging purpose)

## File Manager

Template file format information is located in the following table.

Folder	Description	Template Files
EBS	Create/Update Order Header XML	Refer to <a href="#">Appendix A: Create/Update Order Header Template</a>
EBS	Create/Update Order Line XML	Refer to <a href="#">Appendix B: Create/Update Order Line Template</a>
EBS	Cancel Order/Cancel Line XML	Refer to <a href="#">Appendix C: Cancel Order/Cancel Line Template</a>



## Data Table

### INT\_SYSTEM\_DETAILS

This table contains the information regarding EBS-OM end point.

System	UserName	WebService	SoapEndPoint	RestEndPoint
EBS	<Enter the username here to call the web service endpoint>	<Enter the web service SOAP/REST>	<Enter the SOAP web service endpoint to call the service related to EBS-OM>	<Enter the REST web service endpoint to call the service related to EBS-OM>

### INT\_SYSTEM\_TEMPLATES

This table contains the information regarding templates which are stored in file manager.

System	Operation	Template
EBS	OrderSoap	<Enter the template URL path that is uploaded in File manager>
EBS	lineTemplate	<Enter the template URL path that is uploaded in File manager>
EBS	cancel	<Enter the template URL path that is uploaded in File manager>

## Library Functions

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

### invokeWebService

This library function helps in invoking web service to pass soap Request and return the SOAP response.

```
// Get WebServiceInfo Info from data table and replace it in Template file
//bmql query
resultSet = bmql("Select WebService from INT_SYSTEM_DETAILS where System = $system");
webService = "";
endPoint = "";
//loop through the records
for record in resultSet {
    webService = get(record,"WebService");
}

if (webService == "SOAP") {
//Get SOAP endpoint for the system
resultSet = bmql("Select SoapEndPoint from INT_SYSTEM_DETAILS where System = $system");
endpoint = "";
//loop through the records
for record in resultSet {
    endpoint = get(record,"SoapEndPoint");
    print endpoint;
}
} else {
//Get REST endpoint for the system
resultSet = bmql("Select RestEndPoint from INT_SYSTEM_DETAILS where System = $system");
endpoint = "";
//loop through the records
for record in resultSet {
    endpoint = get(record,"RestEndPoint");
    print endpoint;
}
}

//2. Invoke the web service
headerValues = dict("string");
put(headerValues, "Content-Type", "text/xml; charset=utf-8");

errorString = "Error in "+system+" invocation";
```

```

soapResponse= urldatabypost(endPoint , soapReq ,errorString,headerValues,true); // sends the
soap call and returns response to variable.

print "going to print soapResponse";
print soapResponse;

//3. Return the response
return soapResponse;

```

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

**Commerce BML Library Function Editor: Properties & Parameters**

<b>Name:</b>	<input type="text" value="invokeWebService"/>		<b>#</b>	<b>Parameter Name</b>	<b>Parameter Type</b>
<b>Variable Name:</b>	<input type="text" value="invokeWebService"/>		1	system	String
<b>Description:</b>					
<b>Return Type:</b>	<input type="text" value="String"/>		2	soapReq	String

**Function Editor**

Hide Tools | 
Editor Help

**Attributes**

**Function Wizard**

**Debugger**

**Library Function(s)**

**Main Document Attribute**

**Sub Document Attribute**

**System Attribute**

<b>#</b> <b>Attribute</b>	<b>#</b> <b>Attribute</b>	<b>#</b> <b>Attribute</b>
	transactionLine	
	<b>#</b> <b>Attribute</b>	

Add Attributes

Add Attributes

Add Attributes

## 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";
}

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

return pswd;
```

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

**Commerce BML Library Function Editor: Properties & Parameters**

Name:  # Parameter Name

Variable Name:  # 1 sysName Parameter Type: String

Description:

Return Type:

---

**Function Editor**

Hide Tools | Editor He

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

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

## getTemplateLocation

This library function gets the template location.

```
//1. Get Template File
templateUrl = "";
//bmql query
resultSet = bmql("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];
```

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

The screenshot displays the 'Commerce BML Library Function Editor: Properties & Parameters' window. It shows the function name 'getTemplateLocation', its variable name, and its return type 'String'. A table lists two parameters: 'system' and 'operation', both of type 'String'. Below this, the 'Function Editor' section shows three attribute panels: 'Main Document Attribute' with three attributes (transactionID\_t, orderID, callBackURL), 'Sub Document Attribute' with one attribute (transactionLine), and 'System Attribute' with one attribute (\_system\_user\_first\_name). Each attribute panel has an 'Add Attributes' button at the bottom.

Name:	getTemplateLocation
Variable Name:	getTemplateLocation
Description:	
Return Type:	String

#	Parameter Name	Parameter Type
1	system	String
2	operation	String

Main Document Attribute		
#	Attribute	
1	transactionID_t	↓
2	orderID	↓
3	callBackURL	↓

Sub Document Attribute	
#	Attribute
	transactionLine ↓

System Attribute	
#	Attribute
1	_system_user_first_name ↓

## getOrderRequest

This library function creates the SOAP request for create and update operations.

```
// Get WebserviceInfo Info from data table and replace it in Template file
webservice = "";
resultSet = bmql("Select Webservice from INT_SYSTEM_DETAILS where System = $system");

//loop through the records
for record in resultSet {
    webservice = get(record,"WebService");
}

// check if webservice is soap or rest and choose appropriate template. Right now, its just soap.
HeaderFileLocation="";
if (webservice == "SOAP") {
    operationNew = "OrderSoap";
    HeaderFileLocation = commerce.getTemplateLocation(system, operationNew);
} else {
    operationNew = "OrderRest";
    HeaderFileLocation = commerce.getTemplateLocation(system, operationNew);
}

payload = dict("string");

//Get User Info from data table and replace it in Template file
resultSet = bmql("Select UserName from INT_SYSTEM_DETAILS where System = $system");

//loop through the records to get username and get password.
for record in resultSet {
    userName = get(record,"UserName");
    password = commerce.getPassword(system);
    put(payload,"USERNAME", userName);
    put(payload,"PASSWORD", password);
}

//apply template is called.
```

```

defaultErrorMessage="";
put (payload,"operation",operation);
outFile=applytemplate(HeaderFileLocation,payload,defaultErrorMessage);
lineFileLocation = commerce.getTemplateLocation(system, "lineTemplate");
payload1 = dict("string");
put (payload1,"operation",operation);
lineRequest=applytemplate(lineFileLocation,payload1,defaultErrorMessage);
outFile = replace(outFile ,"$lineInfo$",lineRequest);

//. Return Template File
return outFile;

```

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

**Commerce BML Library Function Editor: Properties & Parameters**

**Name:** getOrderRequest

**Variable Name:** getOrder

**Description:** Common Library Function for creating/updating Order

**Return Type:** String

#	Parameter Name	Parameter Type
1	system	String
2	operation	String

---

**Function Editor**

Hide Tools | Editor Help

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

Main Document Attribute			Sub Document Attribute			System Attribute		
#	Attribute		#	Attribute		#	Attribute	
1	orgID	↓	1	_document_number	↓			
2	paymentTerms_t	↓	2	status_I	↓			
3	priceList	↓	3	lineID	↓			
4	customerID_t	↓	4	_part_number	↓			
5	currency_t	↓	5	requestedQuantity_I	↓			
6	headerID	↓						

transactionLine ↓

Add Attributes

- » Imported Library functions:
  - » getPassword
  - » getTemplateLocation
  - » getOrderDetails



## getOrderDetails

This library function extracts the required information from the SOAP response.

```
status = "";
errorString = "CREATED";
integrationStatus = "Error: Operation failed.";
lineReturn = "";
lineRet = "";
orderNo = "";
dateOrder = "";
//getting xml to find the readxmlmultiple.

startIndex=find (soapResponse, "<OutputParameters");
endIndex=find (soapResponse, "</OutputParameters>");
final = substring(soapResponse, startIndex, endIndex+19);

//xpath for readxmlmultiple to find lineId, returnStatus, HeaderID, orderNumber.

xpath = string[7];
xpath[0] = "/BM_NS:OutputParameters/BM_NS:X_LINE_TBL/BM_NS:P_LINE_TBL_ITEM/BM_NS:LINE_ID";
xpath[1] = "/BM_NS:OutputParameters/BM_NS:X_RETURN_STATUS";
xpath[2] = "/BM_NS:OutputParameters/BM_NS:X_HEADER_REC/BM_NS:HEADER_ID";
xpath[3] = "/BM_NS:OutputParameters/BM_NS:X_HEADER_REC/BM_NS:ORDER_NUMBER";
xpath[4] = "/BM_NS:OutputParameters/BM_NS:X_HEADER_REC/BM_NS:CREATED_BY";
xpath[5] = "/BM_NS:OutputParameters/BM_NS:X_HEADER_REC/BM_NS:ORDERED_DATE";
xpath[6] =
"/BM_NS:OutputParameters/BM_NS:X_MESSAGES/BM_NS:X_MESSAGES_ITEM/BM_NS:MESSAGE_TEXT";

output = readxmlmultiple(final, xpath);
lineID = get(output, xpath[0]);
returnStatus = get(output, xpath[1]);
headerID = get(output, xpath[2]);
orderNum = get(output, xpath[3]);
orderBy = get(output, xpath[4]);
orderDate = get(output, xpath[5]);
if (isnull(orderDate[0])){
print "Orderd Date is null";
}
else{
orderedDate = split(orderDate[0], "T");
dateOrder = orderedDate[0];
```

```

}

messageText = get(output,xpaths[6]);

//Because we do not need to change orderDate and orderBy while updateOperation;
if (status_t=="ORDERED"){
    dateOrder = orderDate_t;
    orderBy[0] = orderPlacedBy_t;
}

//soapResponse for updateOrder returns null.

if(isnull(orderNum[0])){
orderNo = orderNumber;
} else {
orderNo = orderNum[0];
}

// check if returnStatus is success or error.

if(returnStatus[0] == "S") {
status = "ORDERED";
integrationStatus = "Success: Operation Executed.";

} else {
integrationStatus = integrationStatus + " " + messageText[0];
print messageText[0];
status = "CREATED";
dateOrder = datetostr(getdate()) ;
}

//Avoiding Exception Due to "-4712-01-01" in update Order soapResponse in Ordered date element
if (dateOrder == "-4712-01-01") {
dateOrder = datetostr(getdate()) ;
}

```

```

lineDocument = 0;

//returning lineID, status_l to UI
for line in transactionLine{
    if((returnStatus[0] == "S")) {
        if ((line.status_l <> "CANCELED")) {
            lineReturn = lineReturn + line._document_number + "~lineID~";
            lineRet = lineRet + line._document_number + "~status_l~";
            ret = LineID[lineDocument];
            lineReturn = lineReturn + ret + "|";
            lineRet = lineRet + status + "|";
            lineDocument = lineDocument + 1;
        }
    }
    else
    {
        lineReturn = lineReturn + line._document_number + "~lineID~";
        lineRet = lineRet + line._document_number + "~status_l~";
        ret = "Error";
        lineReturn = lineReturn + ret + "|";
        lineRet = lineRet + errorString + "|";
    }
}

returnString =
lineRet+lineReturn+"1~integrationStatusTransaction~"+integrationStatus+"|1~orderNumber~"+order
No+"|1~status_t~"+status+"|1~headerID~"+headerID[0]+"|1~orderPlacedBy_t~"+orderBy[0]+"|1~order
Date_t~"+dateOrder+"|1~soapResponse~"+soapResponse+"|";

return returnString ;

```

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

**Commerce BML Library Function Editor: Properties & Parameters**

Name:  # **Parameter Name** **Parameter Type**

Variable Name:  1 soapResponse String

Description:

Return Type:

---

**Function Editor**

Hide Tools | Editor Help

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

Main Document Attribute			Sub Document Attribute			System Attribute		
#	Attribute		#	Attribute		#	Attribute	
1	orderNumber	↓	transactionLine					
2	status_t	↓	1	_document_number	↓			
3	orderDate_t	↓	2	status_I	↓			
4	orderPlacedBy_t	↓						

## getCancelRequest

This library function builds the SOAP request for cancel order and cancel line.

```
//Get Template File
templateUrl = "";
userName = "";
password = "";
templateFileLocation = commerce.getTemplateLocation(system,"cancel");
payload = dict("string");

//Get User Info from data table and replace it in Template file
resultSet = bqql("Select UserName from INT_SYSTEM_DETAILS where System = $system");

//loop through the records and get password for the webservice
for record in resultSet {
    userName = get(record,"UserName");
    password = commerce.getPassword(system);
    put(payload,"userName", userName);
    put(payload,"password", password);
}
defaultErrorMessage="";
if(operation=="cancelOrder"){
    put(payload,"operation","cancelOrder");
}
else{
    put(payload,"operation","cancelLine");
    for line in transactionLine{
        if(_system_current_document_number==line._document_number){
            put(payload,"lineID",line.lineID);
            put(payload,"cancelledQuantity",string(line.cancelledQuantity));
        }
    }
}

//outfile has the soapRequest for cancel order or cancel line
outfile=applytemplate(templateFileLocation,payload,defaultErrorMessage);

return outfile;
```

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

**Commerce BML Library Function Editor: Properties & Parameters**

Name:

Variable Name:

Description:

Return Type:

#	Parameter Name	Parameter Type
1	system	String
2	operation	String

---

**Function Editor**

Hide Tools | Editor Help

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

Main Document Attribute		Sub Document Attribute		System Attribute	
#	Attribute	#	Attribute	#	Attribute
1	headerID	transactionLine		1	_system_current_document_number
2	rejectExplanation_t	1	lineID		
3	rejectReason_t	2	cancelExplanation		
		3	_document_number		
		4	cancelledQuantity		

- » Imported Library functions:
  - » getPassword
  - » getTemplateLocation

## getDataSecurity

This library function provides the data security elements to soap requests.

```
headerFile_mod=headerFile;
print headerFile_mod;
// 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}}", orgID);
}
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}}", orgID);
}
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}}", warehouseOrgID);
}
//for Order Management. Create/Update/Cancel Order and cancel Line.
if(system == "OM" AND operation == "OM"){
    headerFile_mod= replace(headerFile_mod , "$Responsibility_Name$", "ORDER_MGMT_SUPER_USER");
    headerFile_mod= replace(headerFile_mod , "$Application_Name$", "ONT");
    headerFile_mod= replace(headerFile_mod , "$Security_Group_Name$", "STANDARD");
    headerFile_mod= replace(headerFile_mod , "$Language$", "AMERICAN");
    headerFile_mod= replace(headerFile_mod , "$Org_ID$", orgID);
}
return headerFile_mod;
```





## Actions

### Create Order

The integration is implemented using a single action, "Create Order", as follows:

System Variable Name	Type	Description
N/A		
Variable Name for (Transaction)	Type	Description
N/A		
Variable Name for (Transaction Line)	Type	Description
quantityStatus	String	Availability Status
Imported Commerce Functions		
String getOrderDetails(String soapResponse)		
String getOrder(String system, String operation)		
String invokeWebService(String system, String soapReq)		
String getDataSecurity(String system, String operation, String headerFile)		

---

**IMPORTANT:** The BML for the Create Order Action is located in [Appendix D](#).

---

### Update Order

To show the status of the Order created, the "createAsyncHandler" action is implemented as follows:

System Variable Name	Type	Description
N/A		
Variable Name for (Transaction)	Type	Description
N/A		
Variable Name for (Transaction Line)	Type	Description
N/A		
Imported Commerce Functions		
String getOrderDetails(String soapResponse)		
String getOrder(String system, String operation)		
String invokeWebService(String system, String soapReq)		
String getDataSecurity(String system, String operation, String headerFile)		

---

**IMPORTANT:** The BML for the Update Order Action is located in [Appendix E](#).

---

## Cancel Order

The Cancel order integration is implemented in a single action, "Cancel Order", as follows:

System Variable Name	Type	Description
N/A		
Variable Name for (Transaction)	Type	Description
headerID	String	OrderID in EBS side
Variable Name for (Transaction Line)	Type	Description
_document_number	String	Document Number
requestedQuantity_I	Integer	Quantity of Item Ordered
Imported Commerce Functions		
String getCancelRequest(String system, String operation)		
String invokeWebService(String system, String soapReq)		
String getDataSecurity(String system, String operation, String headerFile)		

**IMPORTANT:** The BML for the Cancel Order Action is located in [Appendix F](#).

## Cancel Line Order (Transaction Line Action)

System Variable Name	Type	Description
_system_current_document_number		
Variable Name for (Transaction)	Type	Description
headerID		
Variable Name for (Transaction Line)	Type	Description
transactionLine	Collection of Sub Documents	
_document_number	String	Document Number
requestedQuantity_I	Integer	Quantity of Item
lineID	String	lineID
Imported Commerce Functions		
String getCancelRequest(String system, String operation)		
String invokeWebService(String system, String soapReq)		
String getDataSecurity(String system, String operation, String headerFile)		

**IMPORTANT:** The BML for the Cancel Line Order Action is located in [Appendix G](#). All actions should be defined for the Advanced Modify – After formula section.

## Step Transitions

Action Variable Name	Available Step	Comments
createOrder (Action : Create Order)	Displayed only in Approved step	Create order is displayed only when the order is approved. If Order is created, step is moved to "Ordered". If Order fails, move to "Failed Order Step"
updateOrder (Action: Update Order)	Ordered	Update Order is displayed only when the order is placed.
cancelOrder (Action: Update Order)	Ordered	Cancel Order is displayed only when the order is placed.
cancelline (Action: Cancel Line)	Ordered	Cancel Line Order is displayed only when the order is placed.
Failed Order (Action: Create Order)	Failed Order	If Order is failed for some reason, Failure message is displayed. Create Order after correcting the error.

Failure

## Formulas

Attribute Name	Formula	Auto Update	Comments
Cancelled Quantity	if( ( Status_I NOT= "CANCELED" ), Quantity,Cancelled Quantity)	Unchecked	In Create Order and Update Order Action; Go to Modify Tab; select Transaction Line Tab; Check Use Formula radio button for Cancelled Quantity.

## Rules

<b>Hiding</b>		
Name: HideQuantity Status: Active		
<b>Condition</b>		
Condition Type: Simple		
<b>Attribute</b>	<b>Operator</b>	<b>Value</b>
Status_t	Equals	CANCELED
Row Grouping: 1		
<b>Action (Attribute to Hide)</b>		
requestedQuantity_I		

<b>Hiding</b>		
Name: HideCancelledQuantity Status: Active		
<b>Condition</b>		
Condition Type: Simple		
<b>Attribute</b>	<b>Operator</b>	<b>Value</b>
Status_t	Not Equals	CANCELED
Row Grouping: 1		
<b>Action (Attribute to Hide)</b>		
cancelledQuantity		



### Sample Payloads

- Refer to [Appendix H: Sample Create Order Payload](#)
- Refer to [Appendix I: Sample Update Order Payload](#)
- Refer to [Appendix J: Cancel Order Payload](#)
- Refer to [Appendix K: Cancel Line Payload](#)

## Appendix A: Create/Update Order Header Template

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">

  <soap:Header
xmlns:ns1="http://xmlns.oracle.com/apps/ont/soapprovider/plsql/oe_inbound_int/">

    <ns1:SOAHeader>

      <ns1:Responsibility>${Responsibility_Name}</ns1:Responsibility>

      <ns1:RespApplication>${Application_Name}</ns1:RespApplication>

      <ns1:SecurityGroup>${Security_Group_Name}</ns1:SecurityGroup>

      <ns1:NLSLanguage>${Language}</ns1:NLSLanguage>

      <ns1:Org_Id>${Org_ID}</ns1:Org_Id>

    </ns1:SOAHeader>

    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
soap:mustUnderstand="1"><wsse:UsernameToken xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns="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></soap:He
ader>

    <soap:Body
xmlns:ns2="http://xmlns.oracle.com/apps/ont/soapprovider/plsql/oe_inbound_int/process_order/"
>

      <ns2:InputParameters>

<ns2:P_API_VERSION_NUMBER>1.0</ns2:P_API_VERSION_NUMBER>

      <ns2:P_INIT_MSG_LIST>T</ns2:P_INIT_MSG_LIST>

      <ns2:P_RETURN_VALUES>T</ns2:P_RETURN_VALUES>

      <ns2:P_ACTION_COMMIT>T</ns2:P_ACTION_COMMIT>

      <ns2:P_HEADER_REC>

        {{# equal operation "UPDATE"}}

        <ns2:HEADER_ID>{{headerID}}</ns2:HEADER_ID>

        {{/equal}}

      <ns2:BOOKED_FLAG>Y</ns2:BOOKED_FLAG>

        <ns2:ORDER_TYPE_ID>1430</ns2:ORDER_TYPE_ID>

        <ns2:ORG_ID>{{orgID}}</ns2:ORG_ID>

        <ns2:PAYMENT_TERM_ID>{{paymentTerms_t}}</ns2:PAYMENT_TERM_ID>

        <ns2:SOLD_TO_ORG_ID>{{customerID_t}}</ns2:SOLD_TO_ORG_ID>

        <ns2:TRANSACTIONAL_CURR_CODE>{{currency_t}}</ns2:TRANSACTIONAL_CURR_CODE>

        <ns2:OPERATION>{{operation}}</ns2:OPERATION>

      </ns2:P_HEADER_REC>

      <ns2:P_LINE_TBL>
```

```
        $lineInfo$
    </ns2:P_LINE_TBL>
    <ns2:P_RTRIM_DATA>n</ns2:P_RTRIM_DATA>
</ns2:InputParameters>
</soap:Body>
</soap:Envelope>
```

## Appendix B: Create/Update Order Line Template

```
{{#each TransactionLine}}
    {{# equal status_l "CANCELED"}}
        {{else}}
            <ns2:P_LINE_TBL_ITEM>
                {{# equal operation "UPDATE"}}
                    <ns2:HEADER_ID>{{headerID}}</ns2:HEADER_ID>
                    {{# equal status_l "ORDERED"}}
                        <ns2:LINE_ID>{{lineID}}</ns2:LINE_ID>
                    {{else}}

<ns2:INVENTORY_ITEM_ID>{{_part_number}}</ns2:INVENTORY_ITEM_ID>
                        <ns2:LINE_TYPE_ID>1427</ns2:LINE_TYPE_ID>
                    {{/equal}}
                    {{else}}

<ns2:INVENTORY_ITEM_ID>{{_part_number}}</ns2:INVENTORY_ITEM_ID>
                        <ns2:LINE_TYPE_ID>1427</ns2:LINE_TYPE_ID>
                    {{/equal}}

<ns2:ORDERED_QUANTITY>{{requestedQuantity_l}}</ns2:ORDERED_QUANTITY>

<ns2:PAYMENT_TERM_ID>{{paymentTerms_t}}</ns2:PAYMENT_TERM_ID>

                {{# equal operation "UPDATE"}}
                    {{# equal status_l "ORDERED"}}
                        <ns2:OPERATION>UPDATE</ns2:OPERATION>

<ns2:SUBSCRIPTION_ENABLE_FLAG>N</ns2:SUBSCRIPTION_ENABLE_FLAG>
                    {{else}}
                        <ns2:OPERATION>CREATE</ns2:OPERATION>
                    {{/equal}}
                {{else}}
                    <ns2:OPERATION>CREATE</ns2:OPERATION>
                {{/equal}}
            </ns2:P_LINE_TBL_ITEM>
        {{/equal}}
    {{/each}}
```



## Appendix C: Cancel Order/Cancel Line Template

```
<soap:Envelope xmlns:wssu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header
xmlns:ns1="http://xmlns.oracle.com/apps/ont/soapprovider/plsql/oe_inbound_int/">
    <ns1:SOAHeader>
      <ns1:Responsibility>$Responsibility_Name$</ns1:Responsibility>
      <ns1:RespApplication>$Application_Name$</ns1:RespApplication>
      <ns1:SecurityGroup>$Security_Group_Name$</ns1:SecurityGroup>
      <ns1:NLSLanguage>$Language$</ns1:NLSLanguage>
      <ns1:Org_Id>$Org_ID$</ns1:Org_Id>
    </ns1:SOAHeader>
    <wsse:Security soap:mustUnderstand="1"
xmlns:env="http://schemas.xmlsoap.org/soap/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>
        <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>
  </soap:Header>
  <soap:Body
xmlns:ns2="http://xmlns.oracle.com/apps/ont/soapprovider/plsql/oe_inbound_int/process_order/"
>
    <ns2:InputParameters>
      <ns2:P_API_VERSION_NUMBER>1.0</ns2:P_API_VERSION_NUMBER>
      <ns2:P_INIT_MSG_LIST>T</ns2:P_INIT_MSG_LIST>
      <ns2:P_RETURN_VALUES>T</ns2:P_RETURN_VALUES>
      <ns2:P_ACTION_COMMIT>T</ns2:P_ACTION_COMMIT>
      {{# equal operation "cancelOrder"}}
      <ns2:P_HEADER_REC>
        <ns2:CANCELLED_FLAG>Y</ns2:CANCELLED_FLAG>
        <ns2:HEADER_ID>{{headerID}}</ns2:HEADER_ID>
        <ns2:OPERATION>UPDATE</ns2:OPERATION>
        <ns2:CHANGE_REASON>{{rejectReason_t}}</ns2:CHANGE_REASON>
        <ns2:CHANGE_COMMENTS>{{rejectExplanation_t}}</ns2:CHANGE_COMMENTS>
      </ns2:P_HEADER_REC>
      {{/equal}}
      <ns2:P_LINE_TBL>
        {{# equal operation "cancelLine"}}
```

```
<ns2:P_LINE_TBL_ITEM>
  <ns2:CANCELLED_FLAG>Y</ns2:CANCELLED_FLAG>
<ns2:CANCELLED_QUANTITY>{{requestedQuantity_1}}</ns2:CANCELLED_QUANTITY>
  <ns2:HEADER_ID>{{headerID}}</ns2:HEADER_ID>
  <ns2:LINE_ID>{{lineID}}</ns2:LINE_ID>
  <ns2:OPERATION>UPDATE</ns2:OPERATION>
  <ns2:SUBSCRIPTION_ENABLE_FLAG>N</ns2:SUBSCRIPTION_ENABLE_FLAG>
</ns2:P_LINE_TBL_ITEM>
  {{/equal}}
</ns2:P_LINE_TBL>
  <ns2:P_RTRIM_DATA>n</ns2:P_RTRIM_DATA>
</ns2:InputParameters>
</soap:Body>
</soap:Envelope>
```

## Appendix D: BML for Create Order Action

```
ret = "";
//get soapRequest
soapRequest = commerce.getOrder("EBS", "CREATE");
//get DataSecurity Element
soapRequest1= commerce.getDataSecurity("OM", "OM", soapRequest);
//get soapresponse
soapResponse = commerce.invokeWebService("EBS", soapRequest1);
startIndex=find (soapResponse, "<OutputParameters");

if(startIndex == -1 ) // This will catch the webservice exceptions
{
return "1~integrationStatusTransaction~ERROR in Webservice Invocation"+
"|1~soapResponse~"+soapResponse ;
}
else{
//get values to return e.g. orderNumber, lineId, orderStatus
ret = commerce.getOrderDetails(soapResponse);
return ret+"1~soapRequest~"+soapRequest1+"|";
}
```

## Appendix E: BML for the Update Order Action

```
//get soapRequest
soapRequest = commerce.getOrder("EBS", "UPDATE");
//get DataSecurity elements
soapRequest1= commerce.getDataSecurity("OM", "OM", soapRequest);
//get soapResponse
soapResponse = commerce.invokeWebService("EBS", soapRequest1);
//get Order Details like lineID, headerId, order Status
startIndex=find (soapResponse, "<OutputParameters");
if(startIndex == -1) // This will catch the webservice exceptions
{
return "1~integrationStatusTransaction~ERROR in Webservice Invocation"+
"|1~soapResponse~"+soapResponse ;
}
else{
//get values to return e.g. orderNumber, lineId, orderStatus
ret = commerce.getOrderDetails(soapResponse);
return ret+"1~soapRequest~"+soapRequest1+"|";
}
```

## Appendix F: BML for the Cancel Order Action

```
status = "";
integrationStatus = "";
//get soapRequest
soapRequest = commerce.getCancelRequest("EBS", "cancelOrder");
//get DataSecurity elements
soapRequest1 = commerce.getDataSecurity("OM", "OM", soapRequest);
//get soapResponse
soapResponse = commerce.invokeWebService("EBS", soapRequest1);

//getting xml out of soapResponse to be worked on readxmlmultiple.

startIndex=find (soapResponse, "<OutputParameters");
endIndex=find (soapResponse, "</OutputParameters>");
final = substring(soapResponse, startIndex,endIndex+19);
xpaths = string[2];
xpaths[0] = "/BM_NS:OutputParameters/BM_NS:X_RETURN_STATUS";
xpaths[1] = "/BM_NS:OutputParameters/BM_NS:X_HEADER_REC/BM_NS:CANCELLED_FLAG";

output = readxmlmultiple(final, xpaths);
returnStatus= get(output,xpaths[0]);
cancelledFlag = get(output,xpaths[1]);
print soapRequest;
if((returnStatus[0] == "S") AND (cancelledFlag[0] == "Y")){
status = "CANCELED";
integrationStatus = "Success: Order cancelled.";
} else {
status = "ORDERED";
integrationStatus = "Error: Order Cancellation failed.";
}
lineRet = "";
lineRetCanQuant = "";
errorString = "ORDERED";
//errorCanQuant = "0";
for line in transactionLine{
    lineRet = lineRet + line._document_number + "~status_1~";
    //lineRetCanQuant = lineRetCanQuant + line._document_number + "~cancelledQuantity~";
    if (status == "CANCELED"){
        lineRet = lineRet + status + "|";
    }
}
```

```
        //lineRetCanQuant = lineRetCanQuant + string(line.requestedQuantity_1) +
"|";
    }
    else{
        lineRet = lineRet + errorString + "|";
        //lineRetCanQuant = lineRetCanQuant + errorCanQuant + "|";

    }

}

return
lineRet+"1~integrationStatusTransaction~"+integrationStatus+"|1~status_t~"+status+"|1~soapRe
sponse~"+soapResponse+"|1~soapRequest~"+soapRequest1+"|";
```

## Appendix G: Cancel Line Order Action

```
status = "";
integrationStatus = "";
//get soap request
soapRequest = commerce.getCancelRequest("EBS", "cancelLine");
//get data Security elements
soapRequest1 = commerce.getDataSecurity("OM", "OM", soapRequest);
//get soap response
soapResponse = commerce.invokeWebService("EBS", soapRequest1);
startIndex=find (soapResponse, "<OutputParameters");
endIndex=find (soapResponse, "</OutputParameters>");
final = substring(soapResponse, startIndex,endIndex+19);
xpaths = string[2];
//Finding Return Status
xpaths[0] = "/BM_NS:OutputParameters/BM_NS:X_RETURN_STATUS";
//Finding cancelled flag
xpaths[1] =
"/BM_NS:OutputParameters/BM_NS:X_LINE_TBL/BM_NS:P_LINE_TBL_ITEM/BM_NS:CANCELLED_FLAG";
output = readxmlmultiple(final, xpaths);
returnStatus= get(output,xpaths[0]);
cancelledFlag = get(output,xpaths[1]);
print soapRequest;
if((returnStatus[0] == "S") AND (cancelledFlag[0] == "Y")){
status = "CANCELED";
integrationStatus ="Success: Line Cancelled";
} else {
status = "ORDERED";
integrationStatus ="Error: Line Cancellation failed.";
}
return _system_current_document_number+"~status_1~"+status+"|" +
_system_current_document_number+"~integrationStatus~"+integrationStatus+"|";
```

## Appendix H: Sample Create Order Payload

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
  <soap:Header
xmlns:ns1="http://xmlns.oracle.com/apps/ont/soapprovider/plsql/oe_inbound_int/">
    <ns1:SOAHeader>
      <ns1:Responsibility>ORDER_MGMT_SUPER_USER</ns1:Responsibility>
      <ns1:RespApplication>ONT</ns1:RespApplication>
      <ns1:SecurityGroup>STANDARD</ns1:SecurityGroup>
      <ns1:NLSLanguage>AMERICAN</ns1:NLSLanguage>
      <ns1:Org_Id>204</ns1:Org_Id>
    </ns1:SOAHeader>
    <wsse:Security soap:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
      <wsse:UsernameToken>
        <wsse:Username>OPERATIONS</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">welcome</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body
xmlns:ns2="http://xmlns.oracle.com/apps/ont/soapprovider/plsql/oe_inbound_int/process_order/"
>
    <ns2:InputParameters>
      <ns2:P_API_VERSION_NUMBER>1.0</ns2:P_API_VERSION_NUMBER>
      <ns2:P_INIT_MSG_LIST>T</ns2:P_INIT_MSG_LIST>
      <ns2:P_RETURN_VALUES>T</ns2:P_RETURN_VALUES>
      <ns2:P_ACTION_COMMIT>T</ns2:P_ACTION_COMMIT>
      <ns2:P_HEADER_REC>
        <ns2:BOOKED_FLAG>Y</ns2:BOOKED_FLAG>
        <ns2:ORDER_TYPE_ID>1430</ns2:ORDER_TYPE_ID>
        <ns2:ORG_ID>204</ns2:ORG_ID>
        <ns2:PAYMENT_TERM_ID>4</ns2:PAYMENT_TERM_ID>
        <ns2:SOLD_TO_ORG_ID>1002</ns2:SOLD_TO_ORG_ID>
        <ns2:TRANSACTIONAL_CURR_CODE>USD</ns2:TRANSACTIONAL_CURR_CODE>
        <ns2:OPERATION>CREATE</ns2:OPERATION>
      </ns2:P_HEADER_REC>
      <ns2:P_LINE_TBL>
        <ns2:P_LINE_TBL_ITEM>
```



```
<ns2:INVENTORY_ITEM_ID>141</ns2:INVENTORY_ITEM_ID>
<ns2:LINE_TYPE_ID>1427</ns2:LINE_TYPE_ID>
<ns2:ORDERED_QUANTITY>1</ns2:ORDERED_QUANTITY>
<ns2:PAYMENT_TERM_ID>4</ns2:PAYMENT_TERM_ID>
<ns2:OPERATION>CREATE</ns2:OPERATION>
</ns2:P_LINE_TBL_ITEM>
</ns2:P_LINE_TBL>
<ns2:P_RTRIM_DATA>n</ns2:P_RTRIM_DATA>
</ns2:InputParameters>
</soap:Body>
</soap:Envelope>
```


## Appendix I: Sample Update Order Payload

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
  <soap:Header
xmlns:ns1="http://xmlns.oracle.com/apps/ont/soapprovider/plsql/oe_inbound_int/">
    <ns1:SOAHeader>
      <ns1:Responsibility>ORDER_MGMT_SUPER_USER</ns1:Responsibility>
      <ns1:RespApplication>ONT</ns1:RespApplication>
      <ns1:SecurityGroup>STANDARD</ns1:SecurityGroup>
      <ns1:NLSLanguage>AMERICAN</ns1:NLSLanguage>
      <ns1:Org_Id>204</ns1:Org_Id>
    </ns1:SOAHeader>
    <wsse:Security soap:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
      <wsse:UsernameToken>
        <wsse:Username>OPERATIONS</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">welcome</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body
xmlns:ns2="http://xmlns.oracle.com/apps/ont/soapprovider/plsql/oe_inbound_int/process_order/"
>
    <ns2:InputParameters>
      <ns2:P_API_VERSION_NUMBER>1.0</ns2:P_API_VERSION_NUMBER>
      <ns2:P_INIT_MSG_LIST>T</ns2:P_INIT_MSG_LIST>
      <ns2:P_RETURN_VALUES>T</ns2:P_RETURN_VALUES>
      <ns2:P_ACTION_COMMIT>T</ns2:P_ACTION_COMMIT>
      <ns2:P_HEADER_REC>
        <ns2:HEADER_ID>358958</ns2:HEADER_ID>
        <ns2:BOOKED_FLAG>Y</ns2:BOOKED_FLAG>
        <ns2:ORDER_TYPE_ID>1430</ns2:ORDER_TYPE_ID>
        <ns2:ORG_ID>204</ns2:ORG_ID>
        <ns2:PAYMENT_TERM_ID>4</ns2:PAYMENT_TERM_ID>
        <ns2:SOLD_TO_ORG_ID>1002</ns2:SOLD_TO_ORG_ID>
        <ns2:TRANSACTIONAL_CURR_CODE>USD</ns2:TRANSACTIONAL_CURR_CODE>
        <ns2:OPERATION>UPDATE</ns2:OPERATION>
      </ns2:P_HEADER_REC>
      <ns2:P_LINE_TBL>
```

```
<ns2:P_LINE_TBL_ITEM>
  <ns2:HEADER_ID>358958</ns2:HEADER_ID>
  <ns2:LINE_ID>672525</ns2:LINE_ID>
  <ns2:ORDERED_QUANTITY>1</ns2:ORDERED_QUANTITY>
  <ns2:PAYMENT_TERM_ID>4</ns2:PAYMENT_TERM_ID>
  <ns2:OPERATION>UPDATE</ns2:OPERATION>
  <ns2:SUBSCRIPTION_ENABLE_FLAG>N</ns2:SUBSCRIPTION_ENABLE_FLAG>
</ns2:P_LINE_TBL_ITEM>
<ns2:P_LINE_TBL_ITEM>
  <ns2:HEADER_ID>358958</ns2:HEADER_ID>
  <ns2:INVENTORY_ITEM_ID>149</ns2:INVENTORY_ITEM_ID>
  <ns2:LINE_TYPE_ID>1427</ns2:LINE_TYPE_ID>
  <ns2:ORDERED_QUANTITY>1</ns2:ORDERED_QUANTITY>
  <ns2:PAYMENT_TERM_ID>4</ns2:PAYMENT_TERM_ID>
  <ns2:OPERATION>CREATE</ns2:OPERATION>
</ns2:P_LINE_TBL_ITEM>
</ns2:P_LINE_TBL>
<ns2:P_RTRIM_DATA>n</ns2:P_RTRIM_DATA>
</ns2:InputParameters>
</soap:Body>
</soap:Envelope>
```

## Appendix J: Cancel Order Payload

```
<soap:Envelope xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header
xmlns:ns1="http://xmlns.oracle.com/apps/ont/soapprovider/plsql/oe_inbound_int/">
    <ns1:SOAHeader>
      <ns1:Responsibility>ORDER_MGMT_SUPER_USER</ns1:Responsibility>
      <ns1:RespApplication>ONT</ns1:RespApplication>
      <ns1:SecurityGroup>STANDARD</ns1:SecurityGroup>
      <ns1:NLSLanguage>AMERICAN</ns1:NLSLanguage>
      <ns1:Org_Id>204</ns1:Org_Id>
    </ns1:SOAHeader>
    <wsse:Security soap:mustUnderstand="1"
xmlns:env="http://schemas.xmlsoap.org/soap/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>
        <wsse:Username>OPERATIONS</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">welcome</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body
xmlns:ns2="http://xmlns.oracle.com/apps/ont/soapprovider/plsql/oe_inbound_int/process_or
der/">
    <ns2:InputParameters>
      <ns2:P_API_VERSION_NUMBER>1.0</ns2:P_API_VERSION_NUMBER>
      <ns2:P_INIT_MSG_LIST>T</ns2:P_INIT_MSG_LIST>
      <ns2:P_RETURN_VALUES>T</ns2:P_RETURN_VALUES>
      <ns2:P_ACTION_COMMIT>T</ns2:P_ACTION_COMMIT>
      <ns2:P_HEADER_REC>
        <ns2:CANCELLED_FLAG>Y</ns2:CANCELLED_FLAG>
        <ns2:HEADER_ID>358958</ns2:HEADER_ID>
        <ns2:OPERATION>UPDATE</ns2:OPERATION>
        <ns2:CHANGE_REASON>Incorrect Qty</ns2:CHANGE_REASON>
        <ns2:CHANGE_COMMENTS/>
      </ns2:P_HEADER_REC>
      <ns2:P_LINE_TBL></ns2:P_LINE_TBL>
      <ns2:P_RTRIM_DATA>n</ns2:P_RTRIM_DATA>
    </ns2:InputParameters>
  </soap:Body>
</soap:Envelope>
```



```
</ns2:InputParameters>  
</soap:Body>  
</soap:Envelope>
```

## Appendix K: Cancel Line Payload

```
<soap:Envelope xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Header xmlns:ns1="http://xmlns.oracle.com/apps/ont/soapprovider/plsql/oe_inbound_int/">
<ns1:SOAHeader>
    <ns1:Responsibility>ORDER_MGMT_SUPER_USER</ns1:Responsibility>
    <ns1:RespApplication>ONT</ns1:RespApplication>
    <ns1:SecurityGroup>STANDARD</ns1:SecurityGroup>
    <ns1:NLSLanguage>AMERICAN</ns1:NLSLanguage>
    <ns1:Org_Id>204</ns1:Org_Id>
</ns1:SOAHeader> <wsse:Security soap:mustUnderstand="1"
xmlns:env="http://schemas.xmlsoap.org/soap/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>OPERATIONS</wsse:Username>
    <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">welcome</wsse:Password>
    </wsse:UsernameToken>
</wsse:Security>
</soap:Header> <soap:Body
xmlns:ns2="http://xmlns.oracle.com/apps/ont/soapprovider/plsql/oe_inbound_int/process_order/"
> <ns2:InputParameters>
    <ns2:P_API_VERSION_NUMBER>1.0</ns2:P_API_VERSION_NUMBER>
    <ns2:P_INIT_MSG_LIST>T</ns2:P_INIT_MSG_LIST>
    <ns2:P_RETURN_VALUES>T</ns2:P_RETURN_VALUES>
    <ns2:P_ACTION_COMMIT>T</ns2:P_ACTION_COMMIT> <ns2:P_LINE_TBL>
<ns2:P_LINE_TBL_ITEM>
<ns2:CANCELLED_FLAG>Y</ns2:CANCELLED_FLAG>
<ns2:CANCELLED_QUANTITY>5</ns2:CANCELLED_QUANTITY>
<ns2:GLOBAL_ATTRIBUTE_CATEGORY />
    <ns2:HEADER_ID>358961</ns2:HEADER_ID>
    <ns2:LINE_ID>672533</ns2:LINE_ID>
<ns2:OPERATION>UPDATE</ns2:OPERATION>
    <ns2:SUBSCRIPTION_ENABLE_FLAG>N</ns2:SUBSCRIPTION_ENABLE_FLAG>
</ns2:P_LINE_TBL_ITEM>
    </ns2:P_LINE_TBL>
    <ns2:P_RTRIM_DATA>n</ns2:P_RTRIM_DATA>
</ns2:InputParameters>
</soap:Body>
</soap:Envelope>
```



Oracle is committed to developing practices and products that help protect the environment

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**

