**Century Salesforce Gateway**

Century Implementation Document

**Legal Disclaimer**

This document contains unpublished, confidential and proprietary information of Excelacom Inc. that is protected by copyright and other intellectual property laws. Excelacom Inc. software and its related documentation are subject to and made available only pursuant to the terms of a license agreement with Excelacom Inc. and may be used or copied only in accordance with the terms of that agreement.

Any use, copying, reproduction, alteration, transmission, or translation of these materials, in whole or in part, in any form or by any means, is strictly prohibited without the prior written permission of Excelacom Inc.

If this material is provided with software licensed by Excelacom Inc. the information herein is provided subject to the terms of the warranty provided with the product license. If this material is not provided with licensed software, the information herein is provided "as is" without warranty of any kind. In either case, there are no other warranties, either expresses or implied, including, but not limited to the implied warranties of merchantability, fitness for a particular purpose, non-infringement, or quality. In no event shall Excelacom Inc. or any of its affiliates be liable for any direct or indirect, special, incidental, consequential, punitive, or exemplary damages arising out of the use or inability to use the materials and/or information contained herein. Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

Excelacom Inc. takes reasonable measures to ensure the quality of the data and other information produced by Excelacom Inc. that is made available herein. However, these materials may contain technical inaccuracies or typographical errors, and are not guaranteed to be error-free. Information may be changed or updated without notice. Excelacom Inc. has no obligation to update the information in this material based on changes to its products or services or those of third parties. Excelacom Inc. may also make improvements or changes to the products or services described in this information at any time without notice.

**Table of Contents**

[1. Objective 5](#_Toc417942180)

[2. Revision History 5](#_Toc417942181)

[3. Scope 5](#_Toc417942182)

[4. Detailed Approach 6](#_Toc417942183)

[5. High Level Process Flow 7](#_Toc417942184)

[6. Technical Overview 7](#_Toc417942185)

[6.1 Removal of Header and Footer in Century Mashup 7](#_Toc417942186)

[6.1.1 Approach. 8](#_Toc417942187)

[6.1.2 Implementation Changes 8](#_Toc417942188)

[6.2 Page Redirection using Century Mashup 10](#_Toc417942189)

[6.2.1 Approach. 10](#_Toc417942190)

[6.2.2 Implementation Changes 11](#_Toc417942191)

[6.3 Save Quote Information and Service Request in External Entity Association Table 15](#_Toc417942192)

[6.3.1 Approach. 16](#_Toc417942193)

[6.3.2 Implementation Changes 16](#_Toc417942194)

[6.4 Notification Calls to Salesforce in Asynchronous mode 26](#_Toc417942195)

[6.4.1 Approach. 26](#_Toc417942196)

[6.4.2 Implementation Changes 26](#_Toc417942197)

[6.5 Page Redirection for Service order Screen: 29](#_Toc417942198)

[6.5.1 Approach. 29](#_Toc417942199)

[6.5.2 Implementation Changes 29](#_Toc417942200)

[6.6 Display the UI task details in Popup: 33](#_Toc417942201)

[6.6.1 Approach. 33](#_Toc417942202)

[6.6.2 Implementation Changes 33](#_Toc417942203)

[6.7 Notification Calls from Century to SFDC 36](#_Toc417942204)

[6.7.1 Approach. 36](#_Toc417942205)

[6.7.2 Implementation Changes 36](#_Toc417942206)

[6.8 Save Opportunity, Account, Contact and Site Details from Salesforce to Century 41](#_Toc417942207)

[6.8.1 Approach. 41](#_Toc417942208)

[6.8.2 Implementation Changes 41](#_Toc417942209)

[6.9 Passing Customer, Account and Contact Details to Customer Manager. 41](#_Toc417942210)

[6.9.1 Approach. 41](#_Toc417942211)

[6.9.2 Implementation Changes 41](#_Toc417942212)

[6.10 Save the Salesforce Information to Century DB. 42](#_Toc417942213)

[6.10.1 Approach. 42](#_Toc417942214)

[6.10.2 Implementation Changes 42](#_Toc417942215)

[6.11 Rules and Validations for Salesforce Information. 42](#_Toc417942216)

[6.11.1 Approach. 42](#_Toc417942217)

[6.11.2 Implementation Changes 42](#_Toc417942218)

[6.12 Status Notification for Salesforce. 43](#_Toc417942219)

[6.12.1 Approach. 43](#_Toc417942220)

[6.12.2 Implementation Changes 43](#_Toc417942221)

[6.13 Send Notification back to Salesforce. 43](#_Toc417942222)

[6.13.1 Approach. 43](#_Toc417942223)

[6.13.2 Implementation Changes 43](#_Toc417942224)

[7. Acronyms and Glossary 44](#_Toc417942225)

# Objective

This document provide the approach for Century Mashup in salesforce to perform the Order fulfillment and Task Completion.

# Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Revision History | | | | | |
| Version | **Date** | **Reason for Change** | **Prepared By** | **Reviewed By** | **Approved By** |
| 0.1 | 21/4/2015 | Century Salesforce Gateway Implementation | Sundarakannan/Vijay | Tamil | Rajasekar |
|  |  |  |  |  |  |

# Scope

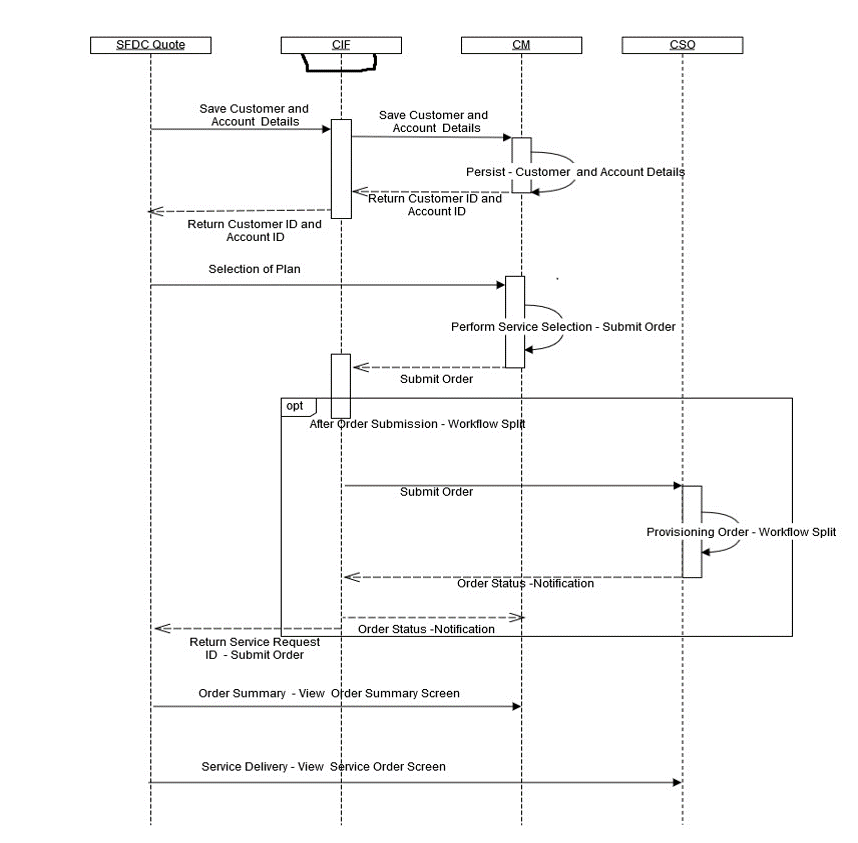
The Scope of these document which includes

* Mashup the Century Screen with Removal of Header and Footer in Salesforce
* End to End Ordering Scenarios using Century Mashup
* Respective Page Redirection using Century Mashup
* Notification Calls from Century to Salesforce.

# Detailed Approach

* Opportunity, Account, Quote and Site information’s (Same like Mandatory Fields of Century) are created in Salesforce and the same is pushed to Century (CM) in the form of XML by selecting the ‘**Select Product’** button.
* Once the Opportunity (Customer), Quote and Account details are pushed to the Century, these data will be processed and persisted in Century and notification will be sent to the Salesforce with the Century Customer Id and Account Id.
* On Successful notification, Salesforce will land the Century Service selection page in a mashup view and user is allowed to select the Century service.
* On Failure notification, Salesforce will throw an Error and user is now allowed to edit and resubmit the data to Century.
* On Success scenario, User will choose the Century Service, Feature, Process and Place an order in Century mashup page.
* Once the Order is placed in Century, the notification will sent to salesforce with service request id and service request status along with Product Pricing Information that is saved in Order Summary Screen.
* If service request status as ‘**In-Progress**’, Update Product Button will be enabled in Salesforce and User is allowed to place Technical and Cancel Supplement using Century Mashup.
* If service request status as ‘**Completed**’, Change/Disconnect Button will be enabled in salesforce and user is allowed to place Change and Disconnect using Century Mashup.
* If service request status as ‘**Held**’, Held Order Button will be enabled in salesforce and user is allowed to place Held order using Century Mashup.
* ‘**Order Summary’** button will be enabled in the Salesforce once the order is placed in Century. On clicking this button, the user is allowed to view Century Order summary in a mashup screen.
* **‘Service Delivery’** button will be enabled in the Salesforce once the order is placed in Century. On clicking this button, the user is allowed to view Century Service delivery in a mashup screen.
* If the user has one SO for the order, then on clicking the “Service Delivery” button in Salesforce will take the user to the Century Task status listing.
* If the user has more than one SO for the order, then on clicking the “Service Delivery” button in Salesforce will take the user to SO listing page
* Once all task are completed in Century Mashup Page, notification will be sent to salesforce with Service Request Status as “Completed”

# High Level Process Flow



# Technical Overview

## Removal of Header and Footer in Century Mashup

* Removal of Header and Footer in Century Mashup of Salesforce which includes the below Screen
  + Service Selection Screen (Which includes Service, Feature, Process and etc.,)
  + Contact

### Approach.

* Parameter are passed as “**hd=true** “in Sale Force to mashup the Century Screen with Removal of Header and Footer.

### Implementation Changes

#### Layout Page.jsp [CM-WEB APP]

* Hiding the Header Panel
* Include the New Script File for Page Redirection

**Code Changes are below**

|  |
| --- |
| <!-- Added for External –CRM Integration – Starts -->  <c:choose>  <c:when test="${hd=='true'}">  <div id="header" style='display:none;'><jsp:include page="Header.jsp"></jsp:include></div>  </c:when>  <c:otherwise>  <div id="header"><jsp:include page="Header.jsp"></jsp:include></div>  </c:otherwise>  </c:choose>  <!-- Added for External –CRM Integration – Ends --> |

* Include the Script file and add hidden parameter for salesforce indication handling

**Code Changes are below**

|  |
| --- |
| <!-- Added for External –CRM Integration – Starts -->  <script type="text/javascript" src="script/modular/externalComp.js"></script>  <!-- Added for External –CRM Integration – Ends -->  <!-- Added for External –CRM Integration – Starts -->  <input type="hidden" value=${hd} id="hdId" />  <input type="hidden" id="hdRedircted" value='${hdRedircted}' />  <input type="hidden" value='${hdParamMap}' id="hdParamMap" />  <!-- Added for External –CRM Integration – Ends --> |

#### LayoutPage.js [CM-WEB APP]

* Hiding the Header Panel

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  if(Ext.get('hdId').getValue() == "true"){  var hdMask = new Ext.LoadMask(Ext.getBody(), {msg:"Loading..."});  hdMask.show();  document.getElementById("mainFrame").onload = function()  { hdMask.hide();}  headerPane.hide(); leftPane.hide(); }  // Added for External –CRM Integration – Ends |

#### BusinessCustomerHome.jsp [CM-WEB APP]

* Include the Script file and add hidden parameter for salesforce indication handling

**Code Changes are below**

|  |
| --- |
| <!-- Added for External –CRM Integration – Starts -->  <script type="text/javascript" src="script/modular/externalComp.js"></script>  <!-- Added for External –CRM Integration – Ends --> |

#### MenuBuilder.java [CM-UI]

* Handling required parameters from salesforce to store in Map Variable if **“hd=true”** (indicates from salesforce)

**Code Changes are below**

|  |
| --- |
| /\* Added for External –CRM Integration – Starts \*/  LOGGER.debug("r.getParameter('sfdcSource'):::::" + r.getParameter("hd"));  requestContext.getFlowScope().put("hd",  Boolean.parseBoolean(r.getParameter("hd")));  requestContext.getFlowScope().put("hdRedircted",  Boolean.parseBoolean(r.getParameter("redircted")));  if (Boolean.parseBoolean(r.getParameter("hd"))) {  Map<String, String[]> paramMap = r.getParameterMap();  String mapString = "{";  for (String key : paramMap.keySet())  mapString += " \"" + key + "\" : \"" + paramMap.get(key)[0]  + "\" , ";  if (mapString.lastIndexOf(",") != -1)  mapString = mapString.substring(0, mapString.lastIndexOf(","));  mapString += "}";  LOGGER.debug (" mapString " + mapString);  requestContext.getFlowScope().put("hdParamMap", mapString);  }  /\* Added for External –CRM Integration – Ends \*/ |

* Assign the Customer Id and Account Id to existing variable if it comes from salesforce parameter.

**Code Changes are below**

|  |
| --- |
| /\* Added for External –CRM Integration – Starts \*/  if (Boolean.parseBoolean(r.getParameter("hd")) && r.getParameter("customerId") != null &&  r.getParameter("entityId") != null && !r.getParameter("customerId").trim().equals("") &&  !r.getParameter("entityId").trim().equals(""))  {  customerId = r.getParameter("customerId");  accountId = r.getParameter("entityId");  }  /\* Added for External –CRM Integration – Ends \*/ |

* Passing the additional required parameter in the URL.

**Code Changes are below**

|  |
| --- |
| /\* Added for External –CRM Integration – Starts \*/ "&isCancelled="+ r.getParameter("IS\_CANCELLED") +"&hd="+Boolean.parseBoolean(r.getParameter("hd"))  +"&redircted="+Boolean.parseBoolean(r.getParameter("redircted")) +"&page="+ r.getParameter("page") +"&hdStatus="+ r.getParameter("hdStatus") );  /\* Added for External –CRM Integration – Ends \*/ |

## Page Redirection using Century Mashup

* Page Redirection using Century Mashup in salesforce which includes the below Service Selection Screen
  + Product Search
  + Order Summary
  + Listing Page
  + Held Order

### Approach.

* Parameter are passed as “**hd=true** “in Sale Force to mashup the Century Screen with Removal of Header and Footer.
* Other Required Parameters which includes Customer Id ,Account Id , Service Request Id , Dates of Order Submission, Held Order Indication and etc.,

### Implementation Changes

#### productSearch.js [SERVORDERMGMT-WEBAPP]

* Parameter are passed as “**hd=true** “ and other Required Parameters which includes Customer Id ,Account Id , Service Request Id , Dates of Order Submission, Held Order Indication and etc., for redirection.

**Code Changes are below**

|  |
| --- |
| //Added for External –CRM Integration – Starts  function formUrlParamFormJson(paramMap) {  if(paramMap == null || paramMap == "")  return "";  var params ="";  for(var key in paramMap)  { params += "&"+key+"="+paramMap[key]; }  return params;  }  //Added for External –CRM Integration – Ends  // Added for External –CRM Integration – Starts  if(parent!=null && parent.document.getElementById('hdId') !=null && parent.document.getElementById('hdId').value=='true' )  {  var hdParamMap = Ext.decode(parent.document.getElementById('hdParamMap').value);  param += formUrlParamFormJson(hdParamMap);  }  // Added for External –CRM Integration – Ends |

* Loading the available service types are default.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  if(parent!=null && parent.document.getElementById('hdId') != null && parent.document.getElementById('hdId').value == 'true' )  loadAvailableServiceType();  // Added for External –CRM Integration – Ends |

#### extservsummary.js [SERVORDERMGMT-WEBAPP]

* Parameter are passed as “**hd=true** “ and other Required Parameters which includes Customer Id ,Account Id , Service Request Id , Dates of Order Submission, Held Order Indication and etc., for redirection.

**Code Changes are below**

|  |
| --- |
| document.forms[0].action = document.forms[0].action + '&\_eventId=retrieveOrderDetails&ordertypeval=3&retriverequestid='+selectedRequestIdList+'&serReqId='+selectedRequestIdList+'&serOrdId='+selectedOrderList+getHDParameters();  //Added for External –CRM Integration – Starts  function formUrlParamFormJson(paramMap)  {  if(paramMap == null || paramMap == "")  return "";  var params ="";  for(var key in paramMap)  {  params += "&"+key+"="+paramMap[key];  }  return params; }  function getHDParameters()  {  var append = "";  if(parent!=null && parent.document.getElementById('hdId') !=null && parent.document.getElementById('hdId').value=='true' )  {  var hdParamMap = Ext.decode(parent.document.getElementById('hdParamMap').value);  append += formUrlParamFormJson(hdParamMap);  }  return append;}  //Added for External –CRM Integration – Ends |

#### ordersummary.js [SERVORDERMGMT-WEBAPP]

* Parameter are passed as “**hd=true** “ and other Required Parameters which includes Customer Id ,Account Id , Service Request Id , Dates of Order Submission, Dates of Order Acceptance for page redirection.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  var externalCompParams=function(){  var append = "";  if(parent!=null && parent.document.getElementById('hdId') != null && parent.document.getElementById('hdId').value == 'true' )  {append += "&hd=true";}  return append; }  // Added for External –CRM Integration – Ends  // Added for External –CRM Integration – Starts  var append = externalCompParams();  window.parent.parent.location.href = "loadServiceRequestIndexPage.exc?parentResourceId=Service\_Order\_Mgmt&childResourceId=ORDER\_TRACKING&currentResourceId=SERVICE\_REQUEST&redirectFlag=Y&orderID="  + redirectOrderID + "&testID=" + testID+append;  // Added for External –CRM Integration – Ends  // Added for External –CRM Integration – Starts  var append = externalCompParams();  window.parent.parent.location = loadUrl + '/loadContactIndexPage.exc?resourceID=CSR\_Menu\_Contact'+append;  // Added for External –CRM Integration – Ends  // Added for External –CRM Integration – Starts  var append = externalCompParams();  window.parent.parent.location = loadUrl + '/loadCustomerIndexPage.exc?reqNAME='+reqName+'&reqID='+reqId+'&reqType='+reqType+'&resourceID=CSR\_Menu\_Customer'+append;  // Added for External –CRM Integration – Ends |

#### loadServOrderManagementPanel.xml [SERVORDERMGMT-WEBAPP]

* Setting the required indication parameters in Scope.

**Code Changes are below**

|  |
| --- |
| <!-- Added for External CRM Integration Starts -->  <set name="flowScope.hd" value="requestParameters.hd" />  <set name="flowScope.hdStatus" value="requestParameters.hdStatus" />  <set name="flowScope.page" value="requestParameters.page" />  <!-- Added for External CRM Integration Ends --> |

#### ordersummary.xml [SERVORDERMGMT-WEBAPP]

* Passing the input value.

**Code Changes are below**

|  |
| --- |
| <!-- Added for External CRM Integration Starts -->  <input name="hd" />  <input name="hdPage" />  <!-- Added for External CRM Integration Ends --> |

#### viewServordermgmt.xml [SERVORDERMGMT-WEBAPP]

* Redirection for Held Order Scenarios.
* Setting the Scope value in input parameter.

**Code Changes are below**

|  |
| --- |
| <!-- Added for External CRM Integration Starts -->  <transition on="${flowScope.out.result=='heldHD'}" to="viewHeldOrder"></transition>  <!-- Added for External CRM Integration Ends -->  <!-- Added for External CRM Integration Starts -->  <input name="hd" value="flowScope.hd" />  <input name="hdPage" value="flowScope.page" />  <!-- Added for External CRM Integration Ends --> |

#### AbstractServOrderController.java [SERVORDERMGMT-UI]

* Load the Order Summary Century Mashup Screen by click on Order summary button in salesforce.
* Load the Held Order Details and then redirect to order summary screen.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  requestContext.getFlowScope().put("processOrder", "false");  // Added for External –CRM Integration – Ends  // Added for External –CRM Integration – Starts  this.externalCompParam();  // Added for External –CRM Integration – Ends  // Added for External –CRM Integration – Starts  /\*\*  \* External Comp Param.  \* @param requestContext the request context  \* @return the map  \*/  public void externalCompParam(RequestContext requestContext, Map<String, String> map){  if(Boolean.parseBoolean(requestContext.getFlowScope().get("hd").toString()) &&  requestContext.getFlowScope().get("page") != null && requestContext.getFlowScope().get("page").toString().equals("ordersummary") )  {  requestContext.getFlowScope().put("processOrder", "true");  map.put(ServiceOrderConst.RESULT,"heldHD"); } }  // Added for External –CRM Integration – Ends |

#### OrderSummaryController.java [SERVORDERMGMT-UI]

* Freezing the screen for s order summary screen.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  if(requestContext.getFlowScope().get("hd") != null && Boolean.parseBoolean(requestContext.getFlowScope().get("hd").toString()) &&  requestContext.getFlowScope().get("hdPage") != null &&  requestContext.getFlowScope().get("hdPage").toString().equals("ordersummary") )  {  selectedProducts.setStatusId(3);  if(httpServletRequest.getParameter("hdStatus") != null)  requestContext.getFlashScope().put("hdStatus", httpServletRequest.getParameter("hdStatus"));  }  // Added for External –CRM Integration – Ends |

#### ServOrderMgmtController.java [SERVORDERMGMT-UI]

* Land to order summary screen based on “**hd=true**” indication in salesforce.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  LOGGER.debug("hierarchyOrderStatus is :: " +hierarchyOrderStatus );  requestContext.getFlowScope().put("hd",  Boolean.parseBoolean(httpServletRequest.getParameter("hd")));  requestContext.getFlowScope().put("page", httpServletRequest.getParameter("page") );  // Added for External –CRM Integration – Ends  // Added for External –CRM Integration – Starts if(Boolean.parseBoolean(requestContext.getFlowScope().get("processOrder").toString()))  {  map.put(ServiceOrderConst.RESULT, ServiceOrderConst.ORDERSUMMARYPAGE);  }  // Added for External –CRM Integration – Ends |

## Save Quote Information and Service Request in External Entity Association Table

* Save the Quote Information and Service Request in External Entity Association Table for first initiation on service request id generation in below screen.
  + Product Search
  + Listing Page

### Approach.

* Quote information will be passed from salesforce and these details will be send as parameter to Century.
* Create of External ID Object for Quote Information and push these object details to Century DB.
* Data will be persist in Century DB for First initiation of service request id generation in Product Search and Listing Page initiation.

### Implementation Changes

#### PartialSaveController.java [SERVORDERMGMT-UI]

* Create an External ID Object from request parameter of Quote information and pass these details to web-service.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  // selectedProducts.setCrmExternalIds(null);  if(Boolean.parseBoolean(httpServletRequest.getParameter("hd")) && httpServletRequest.getParameter("externalQuote") != null )  {  LOGGER.debug("partialSaveDetails selectedProdID :::: " +httpServletRequest.getParameter("selectedProdID"));  LOGGER. debug (" partialSaveDetails externalQuote :::: " +httpServletRequest.getParameter("externalQuote"));  ExternalID externalID = new ExternalID();  externalID.setExternalIdName("Ext\_salesorder\_id"); externalID.setExternalIdValue(httpServletRequest.getParameter("externalQuote") );  List<ExternalID> listExternalID = new ArrayList<ExternalID>();  listExternalID.add(externalID);  ExternalIds externalIds = new ExternalIds();  externalIds.setExternalID(listExternalID);  selectedProducts.setCrmExternalIds(externalIds); servOrderMgmtObj.setOrderInformation(getOrderSummaryHelper().getOrderInfo(httpServletRequest,servOrderMgmtObj.getOrderInformation(),"Sales Object"));  }  // Added for External –CRM Integration – Ends  // Added for External –CRM Integration – Starts  orderDetail = this.getCRMDetails(orderDetail,selectedProducts);  // Added for External –CRM Integration – Ends  // Added for External –CRM Integration – Starts  Private OrderDetail getCRMDetails(OrderDetail orderDetail,SelectedProducts selectedProducts){  LOGGER.debug("Hd :::CRM selectedProducts.getCrmExternalIds() ::: " + selectedProducts.getCrmExternalIds());  if(selectedProducts!=null && selectedProducts.getCrmExternalIds()!=null){  orderDetail.setExternalIds(selectedProducts.getCrmExternalIds());  }  return orderDetail;  }  // Added for External –CRM Integration – Ends |

#### ExtServSummaryController.java [SERVORDERMGMT-UI]

* Getting the Quote information Details from request parameter.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  servOrderMgmtObj = getExtServSummaryCoreController().getCRMOrderDetails(  httpServletRequest,servOrderMgmtObj);  // Added for External –CRM Integration – Ends |

#### ExtServSummaryInter.java [SERVORDERMGMT-UI]

* Interface of CRM Order Details in ExtServSummaryInter.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  public ServOrderMgmtObj getCRMOrderDetails(HttpServletRequest req,ServOrderMgmtObj servOrderMgmtObj);  // Added for External –CRM Integration – Ends |

#### ExtServSummaryCoreController.java [SERVORDERMGMT-UI]

* Create an External ID Object from External Quote Information.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  if(servOrderMgmtObj!=null && servOrderMgmtObj.getProductInformation()!=null)  {  externalIds = this.getCRMExternalDetails(req);  LOGGER.debug(logName +"::"+ ExtServSummaryCoreController.class.getName()+  "::retiriveService externalIds :" + externalIds);  }  // Added for External –CRM Integration – Ends  // Modified for External –CRM Integration  retriveOrderRes = servOrderMgmtImpl.retrieveServiceRequestInfo(  serviceRequestId, soidlist, servOrderMgmtObj,  clearInstanceIdFlag, heldOrderFlag, cancelFlag,  servReqGenFlag, orderStatus, orderType,externalIds, logName);  // Added for External –CRM Integration – Starts  public ServOrderMgmtObj getCRMOrderDetails(HttpServletRequest req,ServOrderMgmtObj servOrderMgmtObj){  LOGGER.debug("hd getCRMOrderDetails :::: "+req.getParameter("hd"));  if(Boolean.parseBoolean(req.getParameter("hd")) && req.getParameter("externalQuote") != null )  {  servOrderMgmtObj.setOrderInformation(orderSummaryHelper.getOrderInfo(  req,servOrderMgmtObj.getOrderInformation(),"Sales Object"));  }  return servOrderMgmtObj;  }  private ExternalIds getCRMExternalDetails(HttpServletRequest req){  LOGGER.debug("hd getCRMExternalDetails :::: "+req.getParameter("hd"));  ExternalIds externalIds = null;  if(Boolean.parseBoolean(req.getParameter("hd")) && req.getParameter("externalQuote") != null )  {  LOGGER.debug("hd externalQuote :::: " +req.getParameter("externalQuote"));  externalIds = new ExternalIds();  ExternalID externalID = new ExternalID();  externalID.setExternalIdName("Ext\_salesorder\_id");  externalID.setExternalIdValue(req.getParameter("externalQuote") );  List<ExternalID> listExternalID = new ArrayList<ExternalID>();  listExternalID.add(externalID);  externalIds.setExternalID(listExternalID);  }  return externalIds;  }  // Added for External –CRM Integration – Ends |

#### OrderSubmissionCoreController.java [SERVORDERMGMT-UI]

* Setting an External ID Object in order detail Information.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  orderDetail = this.getCRMDetails(orderDetail,selectedProducts);  // Added for External –CRM Integration – Ends  // Added for External –CRM Integration – Starts  private OrderDetail getCRMDetails(OrderDetail orderDetail,SelectedProducts selectedProducts){  LOGGER.debug ("Hd :::CRM selectedProducts.getCrmExternalIds() ::: " + selectedProducts.getCrmExternalIds());  if(selectedProducts!=null && selectedProducts.getCrmExternalIds()!=null){  orderDetail.setExternalIds(selectedProducts.getCrmExternalIds());  }  return orderDetail;  }  // Added for External –CRM Integration – Ends |

#### SelectedProducts.java [SERVORDERMGMT-SHARED-WEB]

* Create an External ID Object.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  /\*\* The External Ids bean. \*/  private ExternalIds crmExternalIds;    public ExternalIds getCrmExternalIds() {  return crmExternalIds;  }  public void setCrmExternalIds(ExternalIds crmExternalIds) {  this.crmExternalIds = crmExternalIds;  }  // Added for External –CRM Integration – Ends |

#### ExternalID.java, ExternalIds.java, ServiceOrderConst.java [SERVORDERMGMT-SHARED-WEB]

* Extends Serializable, Cloneable .

**Code Changes are below**

|  |
| --- |
| import java.io.Serializable;  public class ExternalID implements Serializable, Cloneable{  import java.io.Serializable;  public class ExternalIds implements Serializable, Cloneable{  // Added for External –CRM Integration – Starts  /\*\* The ORDERSUMMARY\_LOW. \*/  String ORDERSUMMARY\_LOW = "ordersummary";  // Added for External –CRM Integration – Ends |

#### ServiceOrderConvertor.java [SERVORDERMGMT-BSS-API]

* Create an External ID Object to web service.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  if(orderDetail.getExternalIds() != null && orderDetail.getExternalIds().getExternalID()!=null)  {  Iterator<com.excelacom.century.servordermgmt.shared.bean.ExternalID > externalIDs = orderDetail.getExternalIds().getExternalID().iterator();  com.excelacom.century.servordermgmt.sos.service.webservice.orderdetails.ExternalIds externalIdsWS = new com.excelacom.century.servordermgmt.sos.service.webservice.orderdetails.ExternalIds(); List<com.excelacom.century.servordermgmt.sos.service.webservice.orderdetails.ExternalIds.ExternalID> listWs = new ArrayList<com.excelacom.century.servordermgmt.sos.service.webservice.orderdetails.ExternalIds.ExternalID>();  while(externalIDs.hasNext())  {  com.excelacom.century.servordermgmt.shared.bean.ExternalID externalID = externalIDs.next(); com.excelacom.century.servordermgmt.sos.service.webservice.orderdetails.ExternalIds.ExternalID externalIDWS = new com.excelacom.century.servordermgmt.sos.service.webservice.orderdetails.ExternalIds.ExternalID();  externalIDWS.setExternalIdName(externalID.getExternalIdName() );  externalIDWS.setExternalIdValue(externalID.getExternalIdValue());  listWs.add( externalIDWS);  }  externalIdsWS.setExternalID(listWs);  orderDetailInfo.setExternalIds( externalIdsWS);  }  // Added for External –CRM Integration – Ends |

#### ServOrderMgmtImpl.java [SERVORDERMGMT-BSS-API]

* Setting an External ID Object in response.

**Code Changes are below**

|  |
| --- |
| // Modified for External –CRM Integration  public final OrderDetail retrieveServiceRequestInfo(  final List<String> serviceRequestId, List<String> soList,  ServOrderMgmtObj servOrderMgmtObj, boolean clearInstanceIdFlag,  boolean heldOrderFlag, boolean cancelFlag, boolean servReqGenFlag,  final String orderStatus, final String orderType, ExternalIds externalIds,  final String logName) {  // Added for External –CRM Integration – Starts  if(retriveOrderRes.getSelectedProducts()!=null && externalIds!=null){  LOGGER.debug(logName + ":::" + ServOrderMgmtImpl.class.getName() + ":::"  + " retrieveServiceRequestInfo externalIds :::"+externalIds);  retriveOrderRes.getSelectedProducts().setCrmExternalIds(externalIds);  retriveOrderRes.setExternalIds(externalIds);  }  // Added for External –CRM Integration – Ends |

#### OrderDetailDAO.java [SERVORDERMGMT-SERVICE]

* Saving a Salesforce Information.

**Code Changes are below**

|  |
| --- |
| // Modified for External –CRM Integration – Starts  if(extName!=null && extName.equalsIgnoreCase("Ext\_salesorder\_id")){  pStmt.setString(MagicNumbers.SIX, "SALESFC");  }else {  pStmt.setString(MagicNumbers.SIX, "SFDC");  }  // Modified for External –CRM Integration – Ends |

#### PartialSaveOrderDAOImpl.java [SERVORDERMGMT-SERVICE]

* Saving the External ID Object in Century DB.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  if (saveOrderVo.getOrderDetailInfo().getExternalIds() != null) {  LOGGER. debug ("Order - external id === ");  List<ExternalIDVO> extId = saveOrderVo.getOrderDetailInfo()  .getExternalIds()  .getExternalID();  LOGGER.debug (" saveOrderVo.getServiceRequestId " +saveOrderVo.getServiceRequestId());  for (ExternalIDVO extIdVO : extId) {  if(extIdVO.getExternalIdName()!=null && extIdVO.getExternalIdName().  equalsIgnoreCase("Ext\_salesorder\_id")){  String extName = extIdVO.getExternalIdName();  String extValue = extIdVO.getExternalIdValue();  String ordID = String.valueOf(saveOrderVo.getServiceRequestId());  final PreparedStatementCreator externalOrdIdCreator = this.orderSubmissionStmtCreator.  getExternalIds(OrderSubmissionConstants.INS\_EXTORDID,  extName, extValue, ordID, "102", saveOrderVo);  getJdbcTemplate().update(externalOrdIdCreator);  }  }  }  // Added for External –CRM Integration – Ends |

#### SaveOrderDAOHelper.java [SERVORDERMGMT-SERVICE]

* Handling for Salesforce Option for saving the quote details.

**Code Changes are below**

|  |
| --- |
| // Modified for External CRM Integration  if ((ordersubVo.getApplicationInfo() != null)  && ("OSF".equalsIgnoreCase(ordersubVo  .getApplicationInfo().getProductName()) || "SFDC"  .equalsIgnoreCase(ordersubVo  .getApplicationInfo().getProductName())  || "VFO"  .equalsIgnoreCase(ordersubVo  .getApplicationInfo().getProductName())  || "SALESFC"  .equalsIgnoreCase(ordersubVo  .getApplicationInfo().getProductName()))) { |

#### OrderSubmissionConstants.java [SERVORDERMGMT-SERVICE]

* Constants for external service details.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  String EXTORDID\_COUNT = "select count(1) from entity\_externalid\_assoc where external\_id\_name=? and external\_id\_value=? and entity\_id=? ";  String DEL\_EXTERNAL\_ID\_VAL = "select external\_id\_value from entity\_externalid\_assoc where entity\_id=? and external\_id\_name ='Ext\_salesorder\_id'";  String DEL\_EXTERNAL\_SERV\_STATUS = "select supplement\_type as supplementtype,rel\_serv\_req\_id as relservreqid ,order\_type\_id as ordertypeid,order\_type as ordertype,service\_request\_id as servicerequestid from service\_request where service\_request\_id =?";  String DEL\_EXTERNAL\_ORDER\_STATUS = "select order\_status from service\_request where service\_request\_id =? ";  String DEL\_EXTERNAL\_ORDER\_COUNT = "select count(1) from service\_request where service\_request\_id =? ";  // Added for External –CRM Integration – Ends |

#### SaveOrderConverter.java [SERVORDERMGMT-SERVICE]

* Handling the External Object Information and pass in web service.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  if(orderRequest.getSaveOrderDetail().getOrderDetailInfo()!=null &&  orderRequest.getSaveOrderDetail().getOrderDetailInfo().getExternalIds() != null && orderRequest.getSaveOrderDetail().getOrderDetailInfo().getExternalIds().getExternalId()!=null ){  for (ExternalID extIds : orderRequest.getSaveOrderDetail().getOrderDetailInfo().getExternalIds().getExternalId()) {  if(extIds.getExternalIdName()!=null && extIds.getExternalIdName().  equalsIgnoreCase("Ext\_salesorder\_id")){  saveOrderVo= this.externalCRMDetailsToBS(orderRequest,saveOrderVo);  }  }  // Added for External –CRM Integration – Ends  // Added for External –CRM Integration –  private OrderSubmissionVO externalCRMDetailsToBS(BaseVO baseVO,OrderSubmissionVO saveOrderVo) { if (baseVO != null) {  if (baseVO instanceof PartialSaveRequest) {  PartialSaveRequest orderRequest = (PartialSaveRequest) baseVO;  if(orderRequest.getSaveOrderDetail().getOrderDetailInfo()!=null &&  orderRequest.getSaveOrderDetail().getOrderDetailInfo().getExternalIds() != null)  {  Iterator<ExternalID > externalIDs = orderRequest.getSaveOrderDetail().getOrderDetailInfo().  getExternalIds().getExternalID().iterator();  ExternalIdsVO externalIdsVO = new ExternalIdsVO();  List<ExternalIDVO> listExternalVO = new ArrayList<ExternalIDVO>();  while(externalIDs.hasNext())  {  ExternalID externalID = externalIDs.next();  ExternalIDVO externalIDVO = new ExternalIDVO();  externalIDVO.setExternalIdName(externalID.getExternalIdName() );  externalIDVO.setExternalIdValue(externalID.getExternalIdValue());  listExternalVO.add( externalIDVO);  }  externalIdsVO.setExternalID(listExternalVO);  saveOrderVo.getOrderDetailInfo().setExternalIds(externalIdsVO);  }  LOGGER.debug("partialsave saveOrderVo.getExternalIds() ::::" +saveOrderVo.getOrderDetailInfo().getExternalIds());  } else if (baseVO instanceof SaveOrderRequest) {  SaveOrderRequest orderRequest = (SaveOrderRequest) baseVO;  if(orderRequest.getSaveOrderDetail().getOrderDetailInfo()!=null &&  orderRequest.getSaveOrderDetail().getOrderDetailInfo().getExternalIds() != null) {  Iterator<ExternalID > externalIDs = orderRequest.getSaveOrderDetail().getOrderDetailInfo().  getExternalIds().getExternalID().iterator();  ExternalIdsVO externalIdsVO = new ExternalIdsVO();  List<ExternalIDVO> listExternalVO = new ArrayList<ExternalIDVO>();  while(externalIDs.hasNext())  {  ExternalID externalID = externalIDs.next();  ExternalIDVO externalIDVO = new ExternalIDVO();  externalIDVO.setExternalIdName(externalID.getExternalIdName() );  externalIDVO.setExternalIdValue(externalID.getExternalIdValue());  listExternalVO.add( externalIDVO);  }  externalIdsVO.setExternalID(listExternalVO);  saveOrderVo.getOrderDetailInfo().setExternalIds(externalIdsVO);  }  LOGGER.debug("saveorder saveOrderVo.getExternalIds() ::::" +saveOrderVo.getOrderDetailInfo().getExternalIds());  }  }  return saveOrderVo;  }  //Added for External –CRM Integration – Ends  // Added for External –CRM Integration – Starts  if(orderDtlsVO.getExternalIds()!=null && orderDtlsVO.getExternalIds().getExternalID()!=null ){  ExternalIds externalIdsWS = new ExternalIds();  List<ExternalID> listWs = new ArrayList<ExternalID>();  Iterator<ExternalIDVO > externalIDs1 = orderDtlsVO.getExternalIds().getExternalID().iterator();  while(externalIDs1.hasNext())  {  ExternalIDVO externalIDVO1 = externalIDs1.next();  ExternalID externalIDWS = new ExternalID();  externalIDWS.setExternalIdName(externalIDVO1.getExternalIdName() );  externalIDWS.setExternalIdValue(externalIDVO1.getExternalIdValue());  listWs.add( externalIDWS);  }  externalIdsWS.setExternalID(listWs);  orderDtl.setExternalIds( externalIdsWS);  }  // Added for External –CRM Integration – Ends |

#### SMBOrderSubmissionProcDAOImpl.java [SERVORDERMGMT-SERVICE]

* Saving the External Object Information.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  ordersubVo = getSaveOrderDAOImpl().saveExternalCRMDetail(ordersubVo);  // Added for External –CRM Integration – Ends |

#### SaveOrderDAOImpl.java [SERVORDERMGMT-SERVICE]

* Saving the External Object Details.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  public OrderSubmissionVO saveExternalCRMDetail(  final OrderSubmissionVO ordersubVo) throws CenturyDAOException { try{  if (ordersubVo != null && ordersubVo.getOrderDetailInfo() != null &&  ordersubVo.getOrderDetailInfo().getExternalIds() != null && ordersubVo.getServiceRequestId() != 0) {  List<ExternalIDVO> extId = ordersubVo.getOrderDetailInfo()  .getExternalIds()  .getExternalID();  for (ExternalIDVO extIdVO : extId) {  if(extIdVO.getExternalIdName()!=null && extIdVO.getExternalIdName().  equalsIgnoreCase("Ext\_salesorder\_id")){  String extName = extIdVO.getExternalIdName();  String extValue = extIdVO.getExternalIdValue();  String ordID = String.valueOf(ordersubVo.getServiceRequestId());  long externalIdCount = getJdbcTemplate().queryForLong(  OrderSubmissionConstants.EXTORDID\_COUNT, extName, extValue, ordID);  LOGGER.debug("saveExternalCRMDetail externalIdCount :::" +ordersubVo.getServiceRequestId()+"::"+extName+"::"+extValue);  if(externalIdCount ==0){  final PreparedStatementCreator externalOrdIdCreator = this.orderSubmissionStmtCreator.  getExternalIds(OrderSubmissionConstants.INS\_EXTORDID,  extName, extValue, ordID, "102", ordersubVo);  getJdbcTemplate().update(externalOrdIdCreator);  }  } } }  } catch (DataAccessException dae) {  LOGGER.error("externalCRMDetail Data Access Exception:::::::", dae);  throw new ServiceMgmtDAOException(dae);  } catch (SQLException e) {  LOGGER.error("externalCRMDetail SQl Exception:::::::", e);  throw new ServiceMgmtDAOException(e);  } catch (Exception e) {  LOGGER.error("externalCRMDetail SQl Exception:::::::", e);  }  return ordersubVo;  }  // Added for External –CRM Integration – Ends |

## Notification Calls to Salesforce in Asynchronous mode

* Notification Calls to Salesforce in Asynchronous mode for below scenarios.
  + Partial Save Option
  + Save Order Option
  + Clear Order Option

### Approach.

* Passing the Service Request, Quote and Product Pricing information from Century as asynchronous mode and these information will be passed to salesforce.

### Implementation Changes

#### ServiceOrderManagementServiceImpl.java [SERVORDERMGMT-SERVICE]

* External Component Method will be called as asynchronous mode.

**Code Changes are below**

|  |
| --- |
| **Save Search**  // Added for External –CRM Integration – Starts  ExecutorService executorService = null;  / Added for External –CRM Integration – Starts  executorService = Executors.newSingleThreadExecutor();  try {  final BaseVO saveOrder1 = saveOrder;  executorService.execute(new Runnable() {  @Override  public void run() {  try {  getExternalNotification().externalComp(saveOrder1);    } catch (Exception e) {  LOGGER.error("Exception in inside first catch ::::: " + e);  }  }  });  } catch (Exception e) {  LOGGER.error("Exception inside next catch ::::: " + e);  } finally {  executorService.shutdown();  }  // Added for External –CRM Integration – Ends  **clearOrder**  // Added for External –CRM Integration – Starts  ExecutorService executorService = null;  // Added for External –CRM Integration – Starts  executorService = Executors.newSingleThreadExecutor();  try {  final BaseVO baseVOs= baseVO;  executorService.execute(new Runnable() {  @Override  public void run() {  try {  List<ComponentVO> componentVO = getExternalNotification().getExternalCompDetails(baseVOs);  getExternalNotification().externalCompDelete(componentVO);    } catch (Exception e) {  LOGGER.error("Exception in inside clearOrder catch ::::: " + e);  }  }  });  } catch (Exception e) {  LOGGER.error("Exception inside clearOrder catch ::::: " + e);  } finally {  executorService.shutdown();  }  // Added for External –CRM Integration – Ends  **partialSaveOrder**  // Added for External –CRM Integration – Starts  executorService = Executors.newSingleThreadExecutor();  OrderSubmissionVO saveOrderVo1 = (OrderSubmissionVO) baseVO;  final String strType1= saveOrderVo1.getType();  try {  final BaseVO saveOrder1 = saveOrder;  LOGGER.debug("strType1 executorService before :::"+strType1);  executorService.execute(new Runnable() {  @Override  public void run() {  try {  LOGGER.debug("strType1:::"+strType1);  if (strType1 != null  && strType1  .trim()  .equalsIgnoreCase(  ServiceManagementConstants.SERVICE\_LIST)){  LOGGER.debug("saveOrderVo1.getType() inside :::"+strType1);  getExternalNotification().externalComp(saveOrder1);  }  } catch (Exception e) {  LOGGER.error("Exception in partial save order catch::::: " + e);  }  }  });  } catch (Exception e) {  LOGGER.error("Exception in ::::: " + e);  } finally {  executorService.shutdown();  }  // Added for External –CRM Integration – Ends |

#### Ordersubmission-context.xml [SERVORDERMGMT-SERVICE]

* External Component Configuration calling the CIF Service.

**Code Changes are below**

|  |
| --- |
| <property name="externalNotification" ref="externalServiceNotification"></property>  <!-- Added for External CRM Integration Starts -->  <bean id="externalServiceNotification" class="com.excelacom.century.servordermgmt.cm.sm.servicemgmt.service.businessservice.externalComp.ExternalServiceNotification">  <property name="jdbcTemplate">  <ref bean="jdbcTemplate" />  </property>  <property name="sendExtenalNotification">  <ref bean="sendExtenalNotificationSOAP" />  </property>  </bean>  <jaxws:client id="sendExtenalNotificationSOAP" serviceClass="com.excelacom.century.servordermgmt.cm.sm.servicemgmt.service.businessservice.externalComp.SendExternalNotificationWS"  address="${cif.url}/cif/services/SendExtenalNotificationSOAP">  <jaxws:features>  <bean class="org.apache.cxf.feature.LoggingFeature" />  </jaxws:features>  </jaxws:client>  <!-- Added for External CRM Integration Ends --> |

## Page Redirection for Service order Screen:

* If the user has more than one SO for the order, it will redirect to the SO listing page.
* If the user has only one SO for the order, it will redirect to the SO Task status page.

### Approach.

* Parameter is passed as “**hd=true** “in Sale Force to redirect the Service Order page.
* Other Required Parameters which includes SerReqID, SRFlag.

### Implementation Changes

#### MyOrderHandler.java [OM-JAR]

* Forward to the Task details page if the record list contains one.

**Code changes are below**

|  |
| --- |
| //Added for External -CRM Integration - Starts  final String SerReqID = request.getParameter("SerReqID");  List<String> list = this.getServiceRequest(request, response);  if(list.size() == 1 )  return new ModelAndView("forward:SOTaskDetails.exc?value="+list.get(0)+"&serviceReqID="+SerReqID+"&SRFlag=&hd=true");  //Added for External -CRM Integration - Ends  //Added for External -CRM Integration - Starts  public List<String> getServiceRequest(final HttpServletRequest request,  final HttpServletResponse response) {  LOG.info ( " Boolean.parseBoolean(request.getParameter(hd)) :::: " +Boolean.parseBoolean(request.getParameter("hd"))) ;  if(Boolean.parseBoolean(request.getParameter("hd")))  {  MyOrderHelper myOrderhelper = (MyOrderHelper) getWebApplicationContext().getBean("myOrderHelper");  final String SerReqID = request.getParameter("SerReqID");  LOG.info ( " SerReqID :::: " +SerReqID) ;  List<String> list = myOrderhelper.getServiceRequest(SerReqID);  return list;  }  LOG.info(":::The getServiceRequest in SO :::");  return new ArrayList<String>();  }  //Added for External -CRM Integration - Ends |

#### 6.2.2.2. MyOrderHelper.java [OM-JAR]

* Getting the Service Request ID Details in the List.

**Code Changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  public List<String> getServiceRequest(String serReqID){  return myordImpl.getServiceRequest(serReqID);  }  // Added for External –CRM Integration – Ends |

#### 6.2.2.3. MyOrderImpl.java [OM-JAR]

* Getting the Service Request ID Details List using Service Request Id.

**Code Changes are below**

|  |
| --- |
| //Added for External -CRM Integration - Starts  public List<String> getServiceRequest(String serReqID){  StringBuffer SQL = new StringBuffer();  SQL.append("SELECT s.service\_id ");  SQL.append("FROM service\_request sr, ");  SQL.append(" services s, ");  SQL.append(" entity\_reason\_assoc era, ");  SQL.append(" ui\_set\_critical\_date scd, ");  SQL.append(" namebook nb, ");  SQL.append(" (SELECT a.accountid, ");  SQL.append(" a.vertical\_market\_id, ");  SQL.append(" a.service\_request\_id, ");  SQL.append(" a.accountname, ");  SQL.append(" rlob.description lob, ");  SQL.append(" a.e\_rate ");  SQL.append(" FROM account a, ");  SQL.append(" ref\_lineofbusiness rlob ");  SQL.append(" WHERE a.account\_type\_id = 2 ");  SQL.append(" AND a.lob\_id = rlob.lob\_id ");  SQL.append(" ) acc, ");  SQL.append(" PROJECT\_ENTITY\_ASSOC psa, ");  SQL.append(" project p, ");  SQL.append(" users usr, ");  SQL.append(" addressbook ab1, ");  SQL.append(" addressbook ab2, ");  SQL.append(" (SELECT MAX(uct.next\_contact\_date) AS nexcondate , ");  SQL.append(" service\_request\_id ");  SQL.append(" FROM ui\_cont\_cust\_log uct ");  SQL.append(" GROUP BY service\_request\_id ");  SQL.append(" ) next\_contact\_date, ");  SQL.append(" (SELECT COUNT( local\_biller\_acc\_no) AS local\_biller\_acc\_no1, ");  SQL.append(" LISTAGG (uib.local\_biller\_acc\_no, ', ') WITHIN GROUP ( ");  SQL.append(" ORDER BY ss.serv\_req\_id ) local\_biller\_acc\_no, ");  SQL.append(" ss.serv\_req\_id ");  SQL.append(" FROM UI\_BUILD\_TASKS\_EOD\_DETAILS uib, ");  SQL.append(" services ss ");  SQL.append(" WHERE uib.local\_biller\_name IN ('CSG', 'DDP') ");  SQL.append(" AND uib.service\_order\_id = ss.service\_id ");  SQL.append(" GROUP BY ss.serv\_req\_id ");  SQL.append(" )uib ");  SQL.append("WHERE sr.service\_request\_id = s.serv\_req\_id ");  SQL.append("AND s.so\_level <> 'SOG' ");  SQL.append("AND s.service\_name <> 'OffNet UNI' ");  SQL.append("AND s.service\_id = scd.service\_order\_id(+) ");  SQL.append("AND sr.service\_request\_id = nb.service\_request\_id ");  SQL.append("AND sr.service\_request\_id = acc.service\_request\_id ");  SQL.append("AND s.service\_id =psa.entity\_id (+) ");  SQL.append("AND (psa.entity\_type\_id =105 ");  SQL.append("OR psa.entity\_type\_id IS NULL) ");  SQL.append("AND psa.project\_id = p.project\_id (+) ");  SQL.append("AND p.project\_owner = usr.usr\_login\_id(+) ");  SQL.append("AND s.loc1\_addressid = ab1.addressid(+) ");  SQL.append("AND s.serv\_req\_id = ab1.service\_request\_id(+) ");  SQL.append("AND s.loc2\_addressid = ab2.addressid(+) ");  SQL.append("AND s.serv\_req\_id = ab2.service\_request\_id(+) ");  SQL.append("AND sr.service\_request\_id = next\_contact\_date.service\_request\_id(+) ");  SQL.append("AND sr.service\_request\_id = uib.serv\_req\_id(+) ");  SQL.append("AND s.service\_id = era.entity\_id(+) ");  SQL.append("AND sr.service\_request\_id LIKE ? ");  SQL.append("ORDER BY s.created\_date DESC NULLS LAST");  LOG.info("inside the getServicRequest in MyOrderImpl");  JdbcTemplate jdbcTemplate = new JdbcTemplate(dataSource);  List<String> serviceorderList = jdbcTemplate.queryForList( SQL.toString(), new Object[]{serReqID},String.class);  LOG.info("The List values are getServicRequest:::" + serviceorderList);  return serviceorderList;  //Added for External -CRM Integration - Ends } |

#### 6.2.2.4. MyOrder.js [OM-WEB APP]

* If “**hd=true**”, then Calling the Layout page for getting the session information.
* If “**hd=true**”, then Hiding the back button in SO Screen.
* If “**hd=true**”, then Forward to the Task Detail.

**Code Changes are below**

|  |
| --- |
| //Added for External -CRM Integration - Starts  if(urlParam(location.href,'hd') == 'true' )  {  Ext.Ajax.request({  method: 'POST',  url: 'layoutPage.exc',  success: function( result, request ) {  }  });  }  //Added for External -CRM Integration - Ends  //Added for External -CRM Integration - Starts  if(urlParam(location.href,'hd') == 'true' && document.getElementById('disexecut1e')!=null )  {  document.getElementById('disexecut1e').style.display = 'none';  }  //Added for External -CRM Integration - Ends  //Added for External -CRM Integration - Starts  function urlParam(url,name) {  if (!url) {  url = window.location.href;  }  var results = new RegExp('[\\?&]' + name + '=([^&#]\*)').exec(url);  if (!results) {  return null;  }  return results[1] || null;  }  //Added for External -CRM Integration - Ends  function SOTaskView(c, b) {  var a = document.getElementById("SRFlag").value;  //Added for External -CRM Integration - Starts  var param = "";  if(urlParam(location.href,'hd') == 'true')  param = "&hd=true";  document.frmSearch.action = "SOTaskDetails.exc?value=" + c + "&serviceReqID=" + b + "&SRFlag=" + a+param;  //Added for External -CRM Integration - Ends  document.frmSearch.submit()  } |

## Display the UI task details in Popup:

* Click on the Task Name, UI Task Detail Screen will be shown in Popup.
* Details are loaded in Popup Screen for all status of UI tasks (Completed, In-Progress, Pending, Cancelled and Deferred).
* UI task will be complete by using complete button, popup will be closed and the Service Order-Task Status page is refreshed.
* Back button is removed in popup.

### Approach.

* Parameter are passed as “**hd=true** “in Sale Force to show the UI task page in the Popup.
* Other Required Parameters which includes serviceReqID, SRFlag.

### Implementation Changes

#### ServiceOrderId.jsp [OM-WEB APP]

* UI task detail will be shown in Popup screen.
* Calling the Layout Page to load the session information.

**Code changes are below**

|  |
| --- |
| //Added for External -CRM Integration - Starts  var windowObjectReference;  //Added for External -CRM Integration - Ends  //Added for External -CRM Integration - Starts  if(urlParam(location.href,'hd') == 'true' )  {  Ext.Ajax.request({  method: 'POST',  url: 'layoutPage.exc',  success: function( result, request ) {    }  });  }  //Added for External -CRM Integration - Ends  //Added for External -CRM Integration - Starts  if(urlParam(location.href,'hd') == 'true')  {  var left = (screen.width/2)-(1024/2);  var top = ((screen.height/2)-(550/2))-20;  if(windowObjectReference != null && windowObjectReference !== 'undefined')  {  windowObjectReference.close();  }  url += "&hd=true";  windowObjectReference = window.open(  '',  "Task Details",  "width=1024, height=550,top="+top+", left="+left,  "resizable,scrollbars,status"  );  if(windowObjectReference.document) {  windowObjectReference.document.write('<html><head><title>Task Details</title></head><body height="100%" width="100%" style="margin:3px;padding:0px;overflow:hidden;"><iframe frameborder="0" onLoad="if(window!= null && window.opener!= null)window.opener.taskopen(this,window)" src="' + url + '" height="100%" width="100%" style="border:1px solid #99BBE8;overflow:hidden;margin:0px;padding:0px;"></iframe></body></html>');  }  }  else{  document.catForm.action=url;  document.catForm.submit();  }  }  function urlParam(url,name) {  if (!url) {  url = window.location.href;  }  var results = new RegExp('[\\?&]' + name + '=([^&#]\*)').exec(url);  if (!results) {  return null;  }  return results[1] || null;  }  function detailopen(obj,window)  {  if(window!= null && window.opener!= null)  {    taskopen(obj,window);  } }  function taskopen(control,win)  {  if(control.contentWindow.location.href.indexOf("/OM/SOTaskDetails.exc") != -1 )  {  win.close();  document.getElementById('catForm').submit();  }  }  //Added for External -CRM Integration - Ends    function goPrev()  {  var srflag = '<%=request.getSession().getAttribute("SRFlag")%>';  var serviceRequID = '<%= request.getAttribute("serviceReqID") %>';  Ext.Ajax.request({  url: 'setServiceOrdSession.exc',  success: function(response) {  //Added for External -CRM Integration - Starts  var param = "";  if (srflag != null && srflag !='' && srflag != 'null'){  if(urlParam(location.href,'hd') == 'true')  {  param = '&hd=true';  } document.catForm.action="MyOrder.exc?SRFlag="+srflag+param;//+"&serviceRequID="+serviceRequID;  document.catForm.submit();  }else {  if(urlParam(location.href,'hd') == 'true')  {  param = '?hd=true';  }  document.catForm.action="MyOrder.exc"+param;  document.catForm.submit();  } }  });  }  //Added for External -CRM Integration - Ends |

#### NewScript.jsp [UI Task-WEB APP]

* Removal of Back button in UI task Popup.

**Code changes are below**

|  |
| --- |
| //Added for External -CRM Integration - Starts  function urlParam(url,name) {  if (!url) {  url = window.location.href;  }  var results = new RegExp('[\\?&]' + name + '=([^&#]\*)').exec(url);  if (!results) {  return null;  }  return results[1] || null;  }  //Added for External -CRM Integration - Ends  Ext.onReady(function(){  //Added for External -CRM Integration - Starts  if( parent != null && parent.window.opener != null && urlParam(parent.window.opener.location.href,'hd') == 'true' && document.getElementById('Back') != null)  {  document.getElementById('Back').style.display = 'none';  }  //Added for External -CRM Integration - Ends |

## Notification Calls from Century to SFDC

* If all tasks are completed for that Service Request ID, the Service Request ID status sent notification to salesforce as “Completed”.
* If the order is cancelled for Cancel supplement, the Service Request ID status sent notification to salesforce as “Cancelled”.
* Notification calls are in the Asynchronous mode only.

### Approach.

* Parameter is passed as “**hd=true** “in Sale Force to send notification to salesforce as “Completed” or “Cancelled”.
* Other Required Parameters which includes SerReqID, SRFlag, externalId.

### Implementation Changes

#### Notification-context.xml [notification-service]

* Configure the Salesforce URL and Credential Information from Property file.

**Code Changes are below**

|  |
| --- |
| <!-- Added for External CRM Integration Starts -->  <property name="salesOrderManagementNotification" ref="salesOrderManagementNotification" />  </bean>    <bean id="salesOrderManagementNotification"  class="com.excelacom.notification.service.impl.SalesOrderManagementNotification">  <property name="crmUrl" value="${crm.Url}" />  <property name="crmQuoteUrl" value="${crm.QuoteUrl}" />  <property name="crmgranttype" value="${crm.granttype}" />  <property name="crmclientid" value="${crm.clientid}" />  <property name="crmclientsecret" value="${crm.clientsecret}" />  <property name="crmusername" value="${crm.username}" />  <property name="crmpassword" value="${crm.password}" />    </bean>  <!-- Added for External CRM Integration Ends --> |

#### NotificationDaoSupport.java [notification-service]

* Getting the Service Request ID Status and External Component information from the Century DB.

**Code Changes are below**

|  |
| --- |
| //Added for External –CRM Integration – Starts  public String getExternalComp(String serviceReqId) {  LOGGER.info("Dao getExternalComp. " + serviceReqId);  Object[] args = new Object[] { serviceReqId };  String externalComp =null;  long countExternalComp = getJdbcTemplate().  queryForLong(" select count(1) from ENTITY\_EXTERNALID\_ASSOC where entity\_id='"+serviceReqId+"' and " +  " external\_id\_name='Ext\_order\_id' and created\_by='SFDC'" );  if(countExternalComp>0){  externalComp = (String)getJdbcTemplate().queryForObject("select external\_id\_value from ENTITY\_EXTERNALID\_ASSOC where " + " entity\_id='"+serviceReqId+"' and " +" external\_id\_name='Ext\_order\_id' and created\_by='SFDC'",String.class);  }  LOGGER.info(" External QuoteID getExternalComp 444:::: " +externalComp);  return externalComp;  }  public String getExternalCompServStatus(String serviceReqId) {  LOGGER.info("Dao getExternalCompServStatus. " + serviceReqId);  Object[] args = new Object[] { serviceReqId };  String servRequestStatus =null;  long servRequestCount = getJdbcTemplate().  queryForLong(NotificationQuery.GET\_SERV\_REQUEST\_COUNT,new Object[] { serviceReqId });  if(servRequestCount>0){  servRequestStatus = (String)getJdbcTemplate().queryForObject(NotificationQuery.GET\_SERV\_REQUEST\_STATUS,new Object[] { serviceReqId },String.class);  }  LOGGER.info(" External QuoteID getExternalCompServStatus :::: " +servRequestStatus);  return servRequestStatus;  }  // Added for External –CRM Integration – Ends |

#### NotificationQuery.java [notification-service]

* Constants for getting the Service Request Count and Service Request Status.

**Code Changes are below**

|  |
| --- |
| // Added for CRM Integration -Starts    public static final String GET\_SERV\_REQUEST\_COUNT = "select count(1) from service\_request sr,order\_status os where sr.order\_status\_id = " + " os.order\_status\_id and sr.service\_request\_id = ? ";    public static final String GET\_SERV\_REQUEST\_STATUS = "select os.status\_name from service\_request sr,order\_status os where sr.order\_status\_id = " + " os.order\_status\_id and sr.service\_request\_id = ? ";    // Added for CRM Integration –Ends |

#### NotificationServiceImpl.java [notification-service]

* Service Request ID and Service Request Status will be Send notification.

**Code changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  private SalesOrderManagementNotification salesOrderManagementNotification;  public SalesOrderManagementNotification getSalesOrderManagementNotification() {  return salesOrderManagementNotification;  }    public void setSalesOrderManagementNotification(SalesOrderManagementNotification salesOrderManagementNotification) {  this.salesOrderManagementNotification = salesOrderManagementNotification;  }  // Added for External –CRM Integration – Ends  // Added for External –CRM Integration – Starts  try{  String externalId = notifyHelper.notifyExternalComp(notifyReq);  String servReqStatus = notifyHelper.notifyExternalCompServStatus(notifyReq);  If(externalId!=null && servReqStatus!=null){ salesOrderManagementNotification.notifyToSales(externalId,servReqStatus,notifyReq) ;  }  }  catch (Exception e) {  LOGGER.error("Error in Sales Notification :::", e);  }  break;  // Added for External –CRM Integration – Ends |

#### NotificationHelper.java [notification-service]

* Methods are declared in the interface.

**Code changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  public String notifyExternalComp(NotificationRequest notifyReq);  public String notifyExternalCompServStatus(NotificationRequest notifyReq);  // Added for External –CRM Integration – Ends |

#### NotificationHelperImpl.java [notification-service]

* Getting the External Component and Service Request Status.

**Code changes are below**

|  |
| --- |
| // Added for External –CRM Integration – Starts  public String notifyExternalComp(NotificationRequest notifyReq){  LOGGER.info("Executing notifyExternalComp..");  String externalComp=null;  if(notifyReq!=null && notifyReq.getServiceOrderId()!=null){  String servid = String.valueOf(notifyReq  .getServiceRequestId());  externalComp = notificationDaoSupport.getExternalComp(servid);  LOGGER.info(" External QuoteID notifyExternalComp 33:::: " +externalComp);  }  LOGGER.info("Exiting notifyExternalComp..");  return externalComp;    }  public String notifyExternalCompServStatus(NotificationRequest notifyReq){  LOGGER.info("Executing notifyExternalCompServStatus..");  String servReqStatus=null;  if(notifyReq!=null && notifyReq.getServiceOrderId()!=null){  String servid = String.valueOf(notifyReq  .getServiceRequestId());  servReqStatus = notificationDaoSupport.getExternalCompServStatus(servid);  LOGGER.info(" External QuoteID notifyExternalComp :::: " +servReqStatus);  }  LOGGER.info("Exiting notifyExternalCompServStatus..");  return servReqStatus;  }  // Added for External –CRM Integration – Ends |

#### SalesOrderManagementNotification.java [notification-service]

* External Quote, Order ID and Service Request Status will be formed as JSON String and send these notification to CIF Layer.
* **The newly created file is attached.**



## Save Opportunity, Account, Contact and Site Details from Salesforce to Century

* ‘**salesHttpService’** component which act as consumer and provider for send and receive the information**.**

### Approach.

* Opportunity (Customer), account and site information of Salesforce are received by the component.
* Received inputs are passed to Century in XML format using REST service exposed in Service Integration Module.
* It act as proxy layer. Output send to sales force from Century component as XML which includes Service Status, Century Customer ID and Account ID.
* If fail in century, send response back to sales force as xml with corresponding Error.

### Implementation Changes

* New Component added only for Sales Force Service.



## Passing Customer, Account and Contact Details to Customer Manager.

* ‘salesIntegration’ component added in ServiceMix to expose REST service .

### Approach.

* Redirect the Opportunity (Customer), account and site information of Salesforce to Century CM to persist the details.
* It receive the information inform of xml and redirect to CM with same.

### Implementation Changes

* New Component added only for Sales Force Service.



## Save the Salesforce Information to Century DB.

* ‘sales-ordermgmt-service’ component added in CM for to expose web-service.
* Reads the salesforce information and persist in Century DB.

### Approach.

* Receive the XML format and convert these to BEAN object.
* If XML validation success, then Information will be send to already exposed Century API to perform the Save Operation.
* Bean Validations are handled with SALES-CBF-RULE-UTIL Component.
* Details will be persisted in Century DB and Customer ID and Account ID will be generated.
* Customer ID and Account ID are send back to Salesforce.
* If Validation fails, then Error will be thrown and send back to Salesforce.

### Implementation Changes

* New Component added only for Sales Force to persist the Salesforce details to Century DB.



## Rules and Validations for Salesforce Information.

* **‘sales-cbf-rule-util’** component added in CM for validating data’s from Salesforce.

### Approach.

* Bean Validations are handled for Sale-ordermgmt-Service component.
* Validations such as customer already exists in century, Proper site information, Phone Number validation and etc.,
* If the validation is success, then allow the information to perform save operation.
* If the validation is fail, then error will be thrown to salesforce.

### Implementation Changes

* New Component added only for Sales Force to validate the details.



## Status Notification for Salesforce.

* ‘**SalesNotificationService’** component to expose web-service for forwarding the status information for salesforce.

### Approach.

* Notification details such as Service Request Status, Product Pricing details are send back from CSO or CM.
* Held Information is send back from CM for mapped quote with Service Request ID.
* For “In-Progress” and “Completed” scenario, Notification Details with product pricing are send back from CSO for Mapped Quote with Service Request ID.

### Implementation Changes

* New Component added only for Sales Force to forward the status.



## Send Notification back to Salesforce.

* ‘**sales-notification**’ component added in ServiceMix to sending notification to salesforce.

### Approach.

* Loading the Salesforce Credentials from Property and Receive the Session ID from Salesforce.
* External Quote Information, Order ID, Order Status and Pricing Details will be formed as JSON String.
* These JSON String will be send to Salesforce REST web Service.

### Implementation Changes

* New Component added only for Sales Force to send notification to salesforce as JSON String.



# Acronyms and Glossary

|  |  |
| --- | --- |
| Term/Acronym | Definition |
| CM | Customer Manager |
| CSO | Century Service Orchestrator |
| UI | User Interface |