[Implementation of the hierarchic detail](http://tswiki.tscrm.com/pages/viewpage.action?pageId=43358046)

**Goal**: Implement a [Products] hierarchic detail in an opportunity with a possibility to add a parent and a subordinate record.

**Introduction:** The [Products] section is transformed into a hierarchic section. The following conditions should be implemented in the [Products] detail of an opportunity: whenever a user adds a parent record in the [Product] lookup field, selection of subordinate records should not be permitted. Whenever a user adds subordinate records, multiple selection should be permitted and the following condition should be added: only the subordinate records that are subordinate for the selected parent record product (based on the [Products] section hierarchy) can be added.

**Implementation**

1. Add a [Parent record] lookup field to the opportunity [Product] object, specify [Opportunity product] as an object. Publish.
2. On the OpportunityProductDetailV2 detail page, implement the following:
	1. In the diff schema property:

|  |
| --- |
| {   "operation": "merge",   "name": "DataGrid",   "values": {//includes the hierarchy of the detail list      "hierarchical": true   }}, |

* 1. If an object column has other name than “Parent”, additionally specify the hierarchicalColumnName:”THE NAME OF YOUR COLUMN” property
	2. Create a new button (or edit the existing button) as follows:

|  |
| --- |
| {   "operation": "insert",   "name": "AddRecordButtonV2",   "parentName": "Detail",   "propertyName": "tools",   "index": 0,   "values": {      "itemType": Terrasoft.ViewItemType.BUTTON,      "caption": {"bindTo": "Resources.Strings.AddButtonCaption"},      "controlConfig": {         "menu": {            "items": [               {                  "caption": {"bindTo": "Resources.Strings.AddParentCaption"},                  "click": {bindTo: "addRecord"}               },               {                  "caption": {"bindTo": "Resources.Strings.AddChildCaption"},                  "click": {"bindTo": "addChildRecord"},                  enabled: {bindTo: "getAddChildButtonEnabled"}               }            ]         }      },      "visible": {"bindTo": "getAddRecordButtonVisible"},      "enabled": {"bindTo": "getAddRecordButtonEnabled"}   }} |

* 1. Replace the addGridDataColumns method by adding all necessary columns into it (parent column is required, other columns if needed)

|  |
| --- |
| addGridDataColumns: function (esq) {   this.callParent(arguments);   if (!esq.columns.contains("Opportunity")) {      esq.addColumn("Opportunity");   }   if (!esq.columns.contains("OfferDate")) {      esq.addColumn("OfferDate");   }   if (!esq.columns.contains("Facility")) {      esq.addColumn("Facility");   }   if (!esq.columns.contains("Products")) {      esq.addColumn("Products");   }   if (!esq.columns.contains("Parent")) {      esq.addColumn("Parent");   }}, |

* 1. When adding a subordinate record:

|  |
| --- |
| addChildRecord: function () {   this.set("IsChildRecord", true);   this.sandbox.publish("SaveRecord", {      isSilent: true,      messageTags: [this.sandbox.id]   }, [this.sandbox.id]);}, onCardSaved: function () {   if (this.get("IsChildRecord")) {      this.openProductsLookup();   } else {      this.callParent(arguments);   }}, /\*\* \* Opens the lookup selection page when adding subordinate records \*/openProductsLookup: function () {   var activeRow = this.getActiveRow();   var parentService = null;   if (activeRow === null) {      return;   } else {      parentService = activeRow.get("Products");      if (parentService) {         parentService = parentService.value;      }   }   var opportunityId = activeRow.get("Opportunity").value;    var esq = Ext.create("Terrasoft.EntitySchemaQuery", {      rootSchemaName: "OpportunityProductInterest"   });   esq.addColumn("Id");   esq.addColumn("Products");   esq.filters.add(Terrasoft.createColumnFilterWithParameter(      Terrasoft.ComparisonType.EQUAL, "Opportunity", opportunityId));   esq.getEntityCollection(function (result) {      var existsCollection = [];      if (result.success) {         result.collection.each(function (item) {            existsCollection.push(item.get("Products").value);         }, this);      }      var config = {         entitySchemaName: "Products",         multiSelect: true      };      var filterGroup = Terrasoft.createFilterGroup();      if (existsCollection.length > 0) {         var existsFilter = Terrasoft.createColumnInFilterWithParameters("Id", existsCollection);         existsFilter.comparisonType = Terrasoft.ComparisonType.NOT\_EQUAL;        //You can set up filtering as you see fit         //filterGroup.add("existsFilter", existsFilter);      }      var parentFilter = Terrasoft.createColumnFilterWithParameter(Terrasoft.ComparisonType.EQUAL,         "ParentService", parentService);      filterGroup.add("parentFilter", parentFilter);      config.filters = filterGroup;      this.openLookup(config, this.addCallBack, this);   }, this);}, /\*\* \* callBack function triggered after closing the lookup and selecting some elements. \* @param args \*/addCallBack: function (args) {   var bq = Ext.create("Terrasoft.BatchQuery");   this.selectedRows = args.selectedRows.getItems();   this.selectedRows.forEach(function (item) {      bq.add(this.getInsertQuery(item));   }, this);   if (bq.queries.length) {      bq.execute(this.onItemInserted, this);   }}, /\*\* \* forms a request for adding the selected record as a subordinate one \* @param item \* @returns {Terrasoft.InsertQuery|\*} \*/getInsertQuery: function (item) {   var activeRow = this.getActiveRow();   var recordId = Terrasoft.generateGUID();   var facility = null;   var opportunity = null;   var offerDate = null;   var parentRecordId = activeRow.get("Id");   if (activeRow) {      facility = activeRow.get("Facility");      opportunity = activeRow.get("Opportunity");      offerDate = activeRow.get("OfferDate");   }    var insert = Ext.create("Terrasoft.InsertQuery", {      rootSchemaName: "OpportunityProductInterest"   });//Populating the fields with values from the parent record   insert.setParameterValue("Id", recordId, Terrasoft.DataValueType.GUID);   insert.setParameterValue("Products", item.value, Terrasoft.DataValueType.GUID);   insert.setParameterValue("Opportunity", opportunity.value, Terrasoft.DataValueType.GUID);   insert.setParameterValue("Facility", facility.value, Terrasoft.DataValueType.GUID);   insert.setParameterValue("Quantity", 1, Terrasoft.DataValueType.INTEGER);   insert.setParameterValue("OfferDate", offerDate, Terrasoft.DataValueType.DATE\_TIME);   insert.setParameterValue("Parent", parentRecordId, Terrasoft.DataValueType.GUID);    return insert;}, //callback function triggered after adding all selected records//reloads the gridonItemInserted: function () {     this.reloadGridData();}, |

1. Further, when an added record is opened, a standard opportunity product page will be used. You do not have to modify it.
2. We do not recommend displaying the parent column in the list. The hierarchy will not be built.