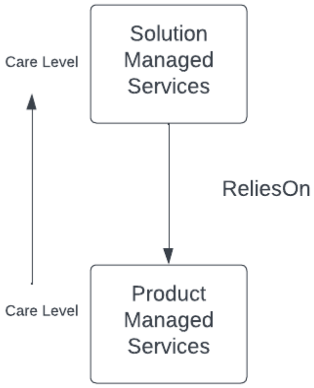
Propagate/Copy Attributes via ReliesOn Relationship

Author: Jeevanprakash Chinya Manjunatha
Technical Architect, Salesforce
Date: 15/June/2023

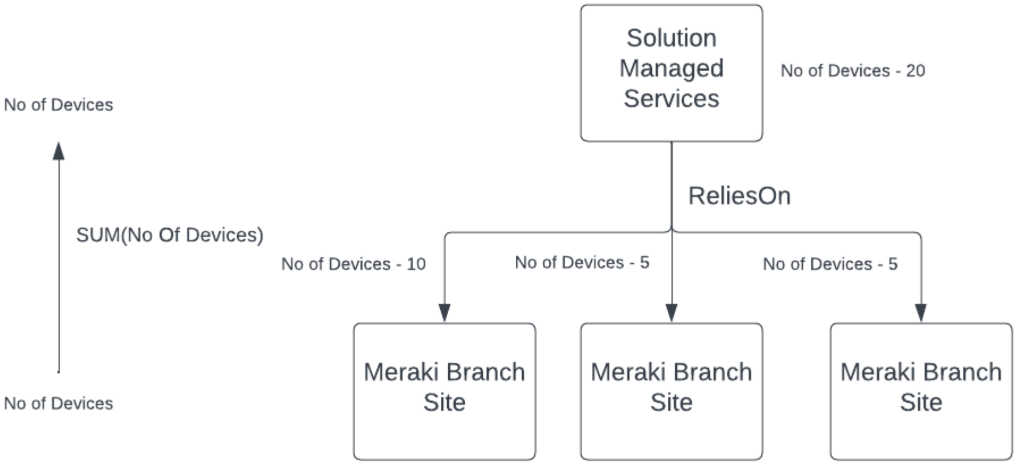
Problem Statement

When products are related via ReliesOn relationships, customers want a capability to propagate attributes from related product instances to the source product instance.
This will help users to not configure the same attributes during cart configuration in the multiple product instances and such attributes might be required to send for downstream systems.

Ex: In below example, Solution Managed Services relieson Product Managed Services and systems needs to propagate Care Level attribute from Product Managed Services to Solution Managed Services.



In below example, Solution Managed Services relieson Meraki Branch Site and systems needs to propagate SUM of No of Devices from all instances.



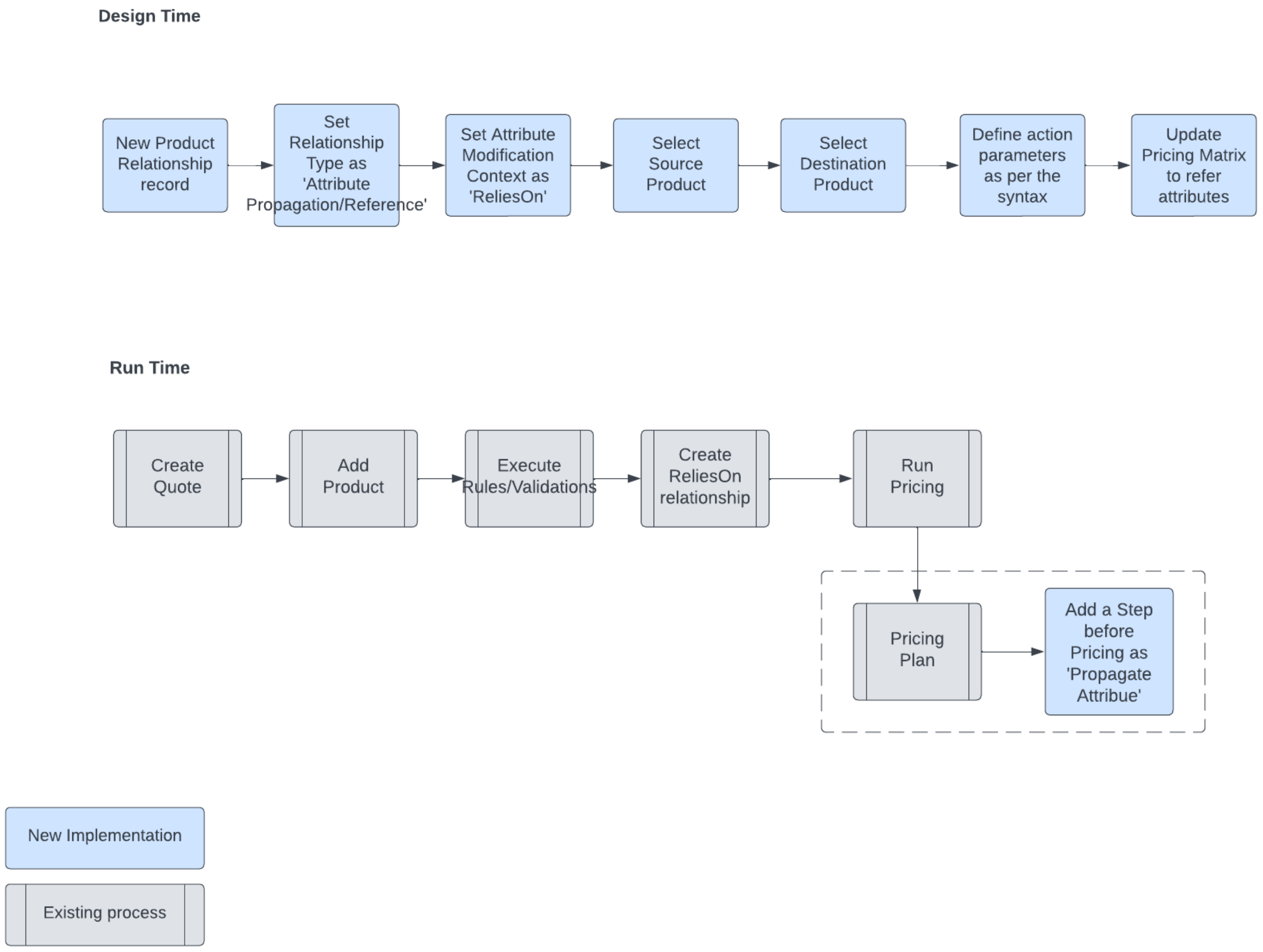
Generally this kind of requirement is achieved using hook implementation but this will be a static solution which needs change or development effort for every new product or change in product.

Technical Design

This feature design will be split into two phases, design time configuration and run time engine.

Design time configuration, which provides ability for product admin to define attributes which need to be propagated.

Run time engine, which refers to the design time definition and current cart product instances and propagate attributes.


Design Time
New Product Relationship record
Set Relationship Type as 'Attribute Propagation/Reference'
Set Attribute Modification Context as 'ReliesOn'
Select Source Product
Select Destination Product
Define action parameters as per the syntax
Update Pricing Matrix to refer attributes
Run Time
Create Quote
Add Product
Execute Rules/Validations
Create ReliesOn relationship
Run Pricing
Pricing Plan
Add a Step before Pricing as 'Propagate Attribue'
New Implementation
Existing process

Design Time

Extend SFI OOTB, Product Relationship object(vlocity_cmt__ProductRelationship__c).

1. Add a new Picklist value for picklist Relationship Type field as ‘Attribute Propagation’
2. Add a new Picklist value for picklist Attribute Modification Context as ‘ReliesOn’

Use the ‘Action Parameter’ field to capture attributes that need to be captured from Related Product Instance to the Source Product Instance.

As part of this example we are supporting direct attribute propagation and also SUM of attribute instance propagation (if relationship is 1:M and depends on attribute data type).

💡 **Tip**
In this example, attribute propagation is done for ReliesOn context only. This can be extended if attributes need to be propagated within the bundle, product hierarchy or cart context. Attribute Modification Context field has OOTB options such as Parent, Bundle and Cart. This could be used as an identifier in the implementation class to fetch attributes from related products. Do an impact analysis to make sure this will not coincide with OOTB implementations.

Syntax

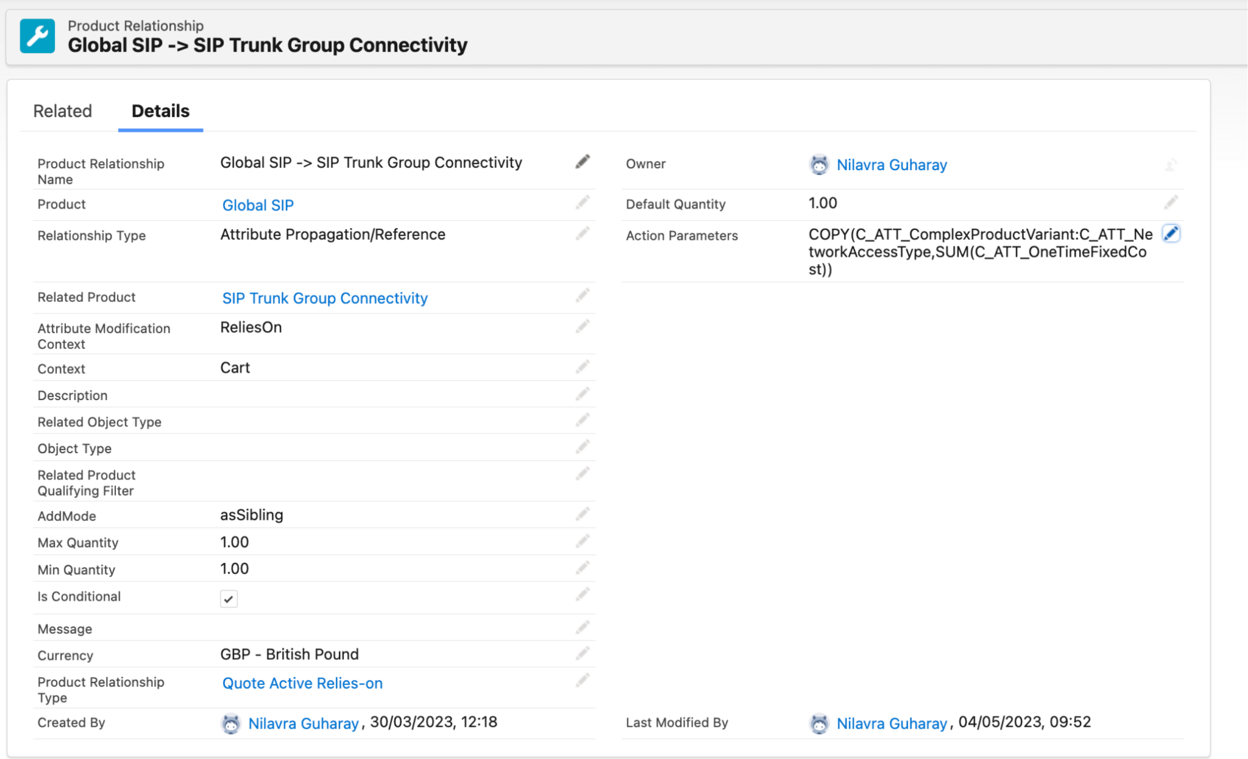
Copy(AttributeCode, SourceAttributeCode:RelatedAttributeCode , SUM(AttributeCode))

AttributeCode - If Source and Related product has same product code which needs to be copied

SourceAttributeCode:RelatedAttributeCode - RelatedAttributeCode from Related product will be copied to SourceAttributeCode of Source product

SUM(AttributeCode) : Attribute code rom Related products will be summed up to AttributeCode of Source product

💡 Tip
In this example, we have done only attribute propagation. Even this can be extended to propagate field values as well. Define unique syntax to differentiate attributes vs fields.



Product Relationship
Global SIP -> SIP Trunk Group Connectivity

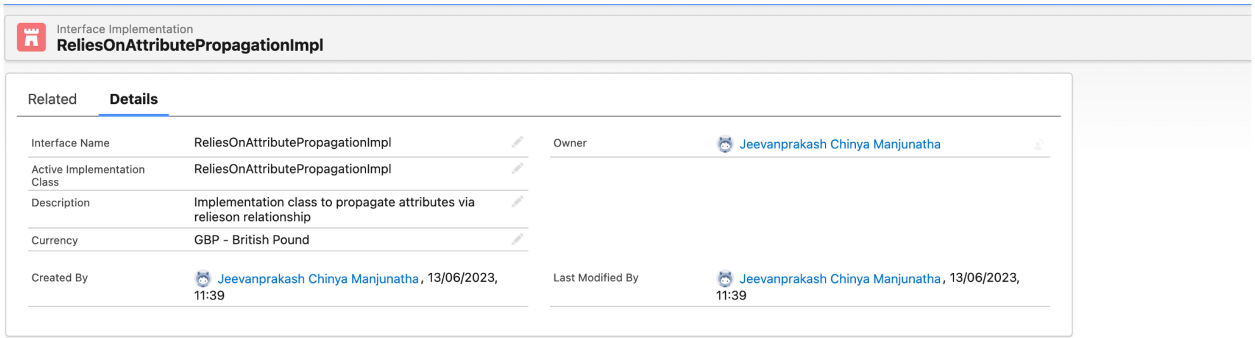
Related | **Details**

Product Relationship Name	Global SIP -> SIP Trunk Group Connectivity	Owner	Nilavra Guharay
Product	Global SIP	Default Quantity	1.00
Relationship Type	Attribute Propagation/Reference	Action Parameters	COPY(C_ATT_ComplexProductVariant:C_ATT_NetworkAccessType,SUM(C_ATT_OneTimeFixedCost))
Related Product	SIP Trunk Group Connectivity		
Attribute Modification Context	ReliesOn		
Context	Cart		
Description			
Related Object Type			
Object Type			
Related Product Qualifying Filter			
AddMode	asSibling		
Max Quantity	1.00		
Min Quantity	1.00		
Is Conditional	☑		
Message			
Currency	GBP - British Pound		
Product Relationship Type	Quote Active Relies-on		
Created By	Nilavra Guharay, 30/03/2023, 12:18	Last Modified By	Nilavra Guharay, 04/05/2023, 09:52

Run Time Engine

Create a new Interface implementation called ***'ReliesOnAttributePropagationImpl'*** which parse created Relies On relationship instances and propagates attribute values based on relies on attribute propagation definition and SUM of same attribute to be propagated based on the relies on attribute propagation definition.

Interface checks for the scope of the invocation either 'Quote' or 'Order', and check respective relationship object for the relies on relationship instance (Quote: vlocity_cmt__QuoteLineItemRelationship__c and Order: vlocity_cmt__OrderItemRelationship__c)



Interface Implementation
ReliesOnAttributePropagationImpl

Interface Name	ReliesOnAttributePropagationImpl	Owner	Jeevanprakash Chinya Manjunatha
Active Implementation Class	ReliesOnAttributePropagationImpl		
Description	Implementation class to propagate attributes via relieson relationship		
Currency	GBP - British Pound		
Created By	Jeevanprakash Chinya Manjunatha, 13/06/2023, 11:39	Last Modified By	Jeevanprakash Chinya Manjunatha, 13/06/2023, 11:39

Interface Implementation can be an invocable class, by passing below I/P.

Inputs:

Scenario 1: Invoked in pricing plan step

invokedFromPricing = true (Required)

Pricing context already has details of lineitems for which pricing is being executed. So no need to pass Id's explicitly.

Scenario 2: Invoked by custom process

invokedFromPricing = false (Required)

ContextId (For Quote its Quote.id and for Order it should be Order.id). When context id is passed, attribute propagation will execute for all the relies on relationship instance exist for that Quote/Order (Bulk execution)

RecordIds (For Quote its list of Quote Line Item Ids and for Order it should be list of Order Item Ids). When RecordIds is passed, attribute propagation will execute only for the relies on relationship instance exist for that lineitem list

Either ContextId or RecordIDs is required if invokedFromPricing is false.

At a time, we need to pass either ContextId or RecordIds as input. Default value for invokedFromPricing is false.

Output:

The class would propagate the attributes from the target to the source and would return the scope (Quote or Order) and the list of Line Item Ids (Order Items or Quote Line Items) for which the attributes were copied to and send back the output in this format

```
{
  "totalSize": 1,
  "messages": [
    {
      "message": "Attributes are updated successfully"
```

```
   }
  ],
  "records": [
   {
     "Scope": "Quote",
     "Records": idslist
   }
  ]
}
```

> 📒 **Note**
> **Parsing attributes from JSON might impact execution time depending on the number of attributes and size of the Quote or Order. It's better to invoke this implementation as an Asynchronous job to handle large Quotes or Orders. In case of invocation in pricing plan step, do an analysis of the Quote/Order size vs execution time and handle pricing accordingly as an batchable process so only set of QLI's/OLI's will be available in the pricing context per batch.**

Code Snippet

```
global with sharing class ReliesOnAttributePropagationImpl implements
vlocity_cmt.VlocityOpenInterface {
    public Boolean invokeMethod(String methodName, Map < String,
Object > input, Map < String, Object > outMap, Map < String, Object >
options) {
        try{
            if (methodName.equals('ReliesOnAttributePropagation')) {
                ReliesOnAttributePropagation(input,outMap, options);
            }
            return true;
        }
        catch(Exception e){
            System.debug('The following exception has occurred: ' +
e.getMessage());
            return null;
        }
    }
    public void ReliesOnAttributePropagation(Map < String, Object >
input, Map < String, Object > outMap, Map < String, Object > options)
```

```
    {

        String scope ;
        String reltypescope;
        Id quoteId ;
        List<Id > lineItemIdlist = new List<Id>();
        Boolean IsInvokedFromPricing =
(Boolean)input.get('invokedFromPricing');
        List<Id>lineItemIdAll = new List<Id>();
        if(IsInvokedFromPricing)
        {
            SObject parent =
(SObject)vlocity_cmt.PricingPlanService.getFromPricingContext('Parent
');

            List<SObject> itemList =
(List<SObject>)vlocity_cmt.PricingPlanService.getFromPricingContext('
LineItemList');

            if(itemList.size()>0)
            {
                for(Sobject sob:itemList)
                {
                    lineItemIdAll.add(sob.id);
                }
            }
            String parentId = (String) parent.get('Id');
            quoteId = (Id)parentId;
            if(parent.getSObjectType() == Quote.sObjectType)
            {

                scope = 'Quote';
            }
            else
            {
                scope = 'Order';
            }

        }
        else
        {
            quoteId = (Id)input.get('ContextId');
            String idlistst =(String)input.get('RecordIds');
            if(idlistst!=null)
            {
                List<string> idlist = idlistst.split(',');

                if(idlist.size()>0)
                {
                    for(string s:idlist)
                    {
                        lineItemIdAll.add((Id)s);
                    }
                    Id lineit = lineItemIdAll[0];

                    List<QuoteLineItem>  qlich= [select id from
quotelineitem where id =: lineit ];

                    if(qlich.size() >0)
                    {
                        scope = 'Quote';
                    }
                    else
                    {
                        scope = 'Order';
                    }

                }
            }

            if(quoteId != null)
            {
                if(quoteId.getSObjectType().getDescribe().getName()
== 'Quote')
                {
                    scope = 'Quote';
                }
                else
                {
                    scope = 'Order';
                }

            }
        }


        if(scope == 'Quote')
        {
```

```
List<vlocity_cmt__QuoteLineItemRelationship__c>existingqprel =
[select id, vlocity_cmt__QuoteLineItemId__c,
vlocity_cmt__RelatedQuoteLineItemId__c,

vlocity_cmt__RelationshipType__c,vlocity_cmt__RelatedAssetReferenceId
__c ,

vlocity_cmt__QuoteLineItemId__r.vlocity_cmt__Product2Id__c,vlocity_cm
t__RelatedQuoteLineItemId__r.vlocity_cmt__Product2Id__c,vlocity_cmt__
ProductRelationshipIdentifier__c FROM
vlocity_cmt__QuoteLineItemRelationship__c

WHERE vlocity_cmt__QuoteId__c =: quoteid OR
vlocity_cmt__QuoteLineItemId__c in:lineItemIdAll];

            set<id> sourceproductid = new set<id>();
            set<id> prprodlist = new set<id>();
            set<string> assetreflist = new set<String>();
            //set<id>relquoteidlist = new set<id>();
            set<id>sourcequoteidlist = new set<id>();
            Map<string,vlocity_cmt__ProductRelationship__c> prrelmap
= new Map<string,vlocity_cmt__ProductRelationship__c>();
            Map<id,string> prreltypemap = new Map<id,string>();

if(vlocity_cmt__QuoteLineItemRelationship__c.sObjectType.getDescribe(
).IsAccessible() )
            {
                if(existingqprel.size()>0)
                {
                    for(vlocity_cmt__QuoteLineItemRelationship__c
qlir : existingqprel)
                    {
                        if(qlir.vlocity_cmt__QuoteLineItemId__c !=
null)
                        {

sourceproductid.add(qlir.vlocity_cmt__QuoteLineItemId__r.vlocity_cmt_
_Product2Id__c);

sourcequoteidlist.add(qlir.vlocity_cmt__QuoteLineItemId__c);
                        }


if(qlir.vlocity_cmt__RelatedAssetReferenceId__c != null)
```

```
                        {
assetreflist.add(qlir.vlocity_cmt__RelatedAssetReferenceId__c );
                        }


                    }
                }

                Map<String,QuoteLineItem > qlimap = new
Map<String,QuoteLineItem >();
                Map<id,QuoteLineItem > qlisourcemap = new
Map<id,QuoteLineItem >([select
id,vlocity_cmt__AttributeSelectedValues__c,vlocity_cmt__Product2Id__c
from QuoteLineItem where id in:sourcequoteidlist]);
                // sorted by createddate so the latest instance will
be stored in the map
                for(QuoteLineItem qli:[select
id,vlocity_cmt__AttributeSelectedValues__c,vlocity_cmt__Product2Id__c
,vlocity_cmt__AssetReferenceId__c from QuoteLineItem where
vlocity_cmt__AssetReferenceId__c in:assetreflist order by createddate
])
                {
qlimap.put(qli.vlocity_cmt__AssetReferenceId__c,qli);
                }
                Map<string,Asset> assetmap = new Map<string,Asset>();
                for(asset a:[select
vlocity_cmt__AssetReferenceId__c,id,
vlocity_cmt__AttributeSelectedValues__c,Product2Id  from asset where
vlocity_cmt__AssetReferenceId__c in:assetreflist order by createddate
])
                {
assetmap.put(a.vlocity_cmt__AssetReferenceId__c,a);
                }

                Map<string,OrderItem> oimap = new
Map<string,OrderItem>();
                for(OrderItem o:[select id,
vlocity_cmt__AttributeSelectedValues__c,vlocity_cmt__AssetReferenceId
__c,vlocity_cmt__Product2Id__c  from orderitem where
vlocity_cmt__AssetReferenceId__c in:assetreflist order by createddate
])
                {
                    oimap.put(o.vlocity_cmt__AssetReferenceId__c,o);
                }


if(vlocity_cmt__ProductRelationship__c.sObjectType.getDescribe().IsAc
cessible()){

                    for (vlocity_cmt__ProductRelationship__c   prList
: [SELECT Id,Name,Product_Relationship_Type__r.Name,
vlocity_cmt__Product2Id__c,vlocity_cmt__AttributeModificationContext_
_c,

vlocity_cmt__RelatedProductId__c,vlocity_cmt__RelationshipType__c,vlo
city_cmt__ActionParameters__c ,

Product_Relationship_Type__r.vlocity_cmt__RelationshipTypeScope__c,Pr
oduct_Relationship_Type__c

FROM vlocity_cmt__ProductRelationship__c

WHERE vlocity_cmt__Product2Id__c in :sourceproductid])
                    {

prreltypemap.put(prList.id,prList.Product_Relationship_Type__r.Name);
                        if(prList.vlocity_cmt__RelationshipType__c ==
'Attribute Propagation/Reference' &&
prList.vlocity_cmt__AttributeModificationContext__c =='ReliesOn' &&
prList.vlocity_cmt__ActionParameters__c!=null)
                        {

                            String s =
prList.vlocity_cmt__Product2Id__c + ':' +
prList.vlocity_cmt__RelatedProductId__c ;


prprodlist.add(prList.vlocity_cmt__Product2Id__c);
                            prrelmap.put(s,prList);
                        }

                    }
                }
                List<Id> qlilistup = new List<id>();
                Map<id,QuoteLineItem> qliattrimap = new
Map<id,QuoteLineItem>();
                if(prprodlist.size()>0)
                {
                    for(vlocity_cmt__QuoteLineItemRelationship__c
qlir :existingqprel)
                    {


if(prprodlist.contains(qlir.vlocity_cmt__QuoteLineItemId__r.vlocity_c
mt__Product2Id__c))
                        {
                            QuoteLineItem qlsource =
qlisourcemap.get(qlir.vlocity_cmt__QuoteLineItemId__c);


if(qlir.vlocity_cmt__RelatedAssetReferenceId__c != null)
                            {

if(oimap.containsKey(qlir.vlocity_cmt__RelatedAssetReferenceId__c ))
                                {
                                    String reltype =
prreltypemap.get(qlir.vlocity_cmt__ProductRelationshipIdentifier__c);
                                    OrderItem oi =
oimap.get(qlir.vlocity_cmt__RelatedAssetReferenceId__c);
                                    String s =
qlsource.vlocity_cmt__Product2Id__c + ':' +
oi.vlocity_cmt__Product2Id__c;
                                    if(prrelmap.containsKey(s))
                                    {

vlocity_cmt__ProductRelationship__c Prrel = prrelmap.get(s);

                                        string sumcopy;
                                        string copyatt;
                                        string totalparam =
actionparamatersmanipulate(Prrel.vlocity_cmt__ActionParameters__c);
                                        copyatt =
totalparam.split(';')[0];
                                        sumcopy =
totalparam.split(';')[1];
                                        Integer k=0;

if(qliattrimap.containskey(qlsource.id))
                                        {
                                            k=1;
                                        }
```

```
                                         String a = 
modifyAttributes(qlsource.vlocity_cmt__AttributeSelectedValues__c,oi.
vlocity_cmt__AttributeSelectedValues__c,copyatt,sumcopy,k);

                                         if(a!=null)
                                         {

qlsource.vlocity_cmt__AttributeSelectedValues__c = a;
                                         }

                                         qlilistup.add(qlsource.id);

qliattrimap.put(qlsource.id,qlsource);

                                     }
                                 }
                                 else 
if(qlimap.containsKey(qlir.vlocity_cmt__RelatedAssetReferenceId__c))
                                 {

                                     QuoteLineItem qlrel = 
qlimap.get(qlir.vlocity_cmt__RelatedAssetReferenceId__c);

                                     String s = 
qlsource.vlocity_cmt__Product2Id__c + ':' + 
qlrel.vlocity_cmt__Product2Id__c ;
                                     if(prrelmap.containsKey(s))
                                     {


vlocity_cmt__ProductRelationship__c Prrel = prrelmap.get(s);
                                         string sumcopy;
                                         string copyatt;
                                         string totalparam = 
actionparamatersmanipulate(Prrel.vlocity_cmt__ActionParameters__c);
                                         copyatt = 
totalparam.split(';')[0];
                                         sumcopy = 
totalparam.split(';')[1];
                                         Integer k=0;

if(qliattrimap.containskey(qlsource.id))
                                         {
                                             k=1;
                                         }
```

```
                                        String a =
modifyAttributes(qlsource.vlocity_cmt__AttributeSelectedValues__c,qlr
el.vlocity_cmt__AttributeSelectedValues__c,copyatt,sumcopy,k);

                                        if(a!=null)
                                        {

qlsource.vlocity_cmt__AttributeSelectedValues__c = a;
                                        }
                                        qlilistup.add(qlsource.id);

qliattrimap.put(qlsource.id,qlsource);

                                    }
                                }

                                else
if(assetmap.containsKey(qlir.vlocity_cmt__RelatedAssetReferenceId__c)
)
                                {
                                    asset a =
assetmap.get(qlir.vlocity_cmt__RelatedAssetReferenceId__c);
                                    String reltype =
prreltypemap.get(qlir.vlocity_cmt__ProductRelationshipIdentifier__c);
                                    String s =
qlsource.vlocity_cmt__Product2Id__c + ':' + a.Product2Id ;
                                    if(prrelmap.containsKey(s))
                                    {

vlocity_cmt__ProductRelationship__c Prrel = prrelmap.get(s);
                                        string sumcopy;
                                        string copyatt;
                                        string totalparam =
actionparamatersmanipulate(Prrel.vlocity_cmt__ActionParameters__c);
                                        copyatt =
totalparam.split(';')[0];

                                        sumcopy =
totalparam.split(';')[1];
                                        Integer k=0;

if(qliattrimap.containskey(qlsource.id))
                                        {
                                            k=1;
                                        }
                                        String ax = 
modifyAttributes(qlsource.vlocity_cmt__AttributeSelectedValues__c,a.v
locity_cmt__AttributeSelectedValues__c,copyatt,sumcopy,k);
                                        if(ax!=null)
                                        {

qlsource.vlocity_cmt__AttributeSelectedValues__c = ax;
                                        }
                                        qlilistup.add(qlsource.id);

qliattrimap.put(qlsource.id,qlsource);

                                    }

                                }
                            }

                        }
                    }
                }

                Map<String,Object> outputmap = new 
Map<String,Object>();
                Map<String,Object> theObj = new Map<String, 
Object>();
                theObj.put('Scope',Scope);
                theObj.put('Records',qlilistup);

                Map<String,Object> theObj1 = new Map<String, 
Object>();
                theObj1.put('message','Attributes are updated 
successfully');
                List<Object>msglist =new List<Object>();
                msgList.add(theObj1);
                List<Object>recList =new List<Object>();
                recList.add(theObj);

                outputmap.put('totalSize', 1);
                outputmap.put('messages',msgList);
                outputmap.put('records',reclist);

                if(qliattrimap.size()>0)
                {
                    Database.update(qliattrimap.values());
                    outMap.put('Creation
Status',json.serialize(outputmap));
                }
            }
        }
        else if(Scope == 'Order')
        {
            List<vlocity_cmt__OrderItemRelationship__c >existingqprel
= [select id, vlocity_cmt__OrderItemId__c ,
vlocity_cmt__RelatedOrderItemId__c ,

vlocity_cmt__RelationshipType__c,vlocity_cmt__RelatedAssetReferenceId
__c ,

vlocity_cmt__OrderItemId__r.vlocity_cmt__Product2Id__c,vlocity_cmt__R
elatedOrderItemId__r.vlocity_cmt__Product2Id__c,vlocity_cmt__ProductR
elationshipIdentifier__c FROM vlocity_cmt__OrderItemRelationship__c
WHERE vlocity_cmt__OrderId__c  =: quoteid

or vlocity_cmt__OrderItemId__c in:lineItemidall];
            set<id> sourceproductid = new set<id>();
//sourceproductid
            set<id> prprodlist = new set<id>();
            set<string> assetreflist = new set<String>();
            set<id>relorderidlist = new set<id>();
            set<id>sourceorderidlist = new set<id>();
            Map<string,vlocity_cmt__ProductRelationship__c> prrelmap
= new Map<string,vlocity_cmt__ProductRelationship__c>();
            Map<id,string> prreltypemap = new Map<id,string>();

if(vlocity_cmt__OrderItemRelationship__c.sObjectType.getDescribe().Is
Accessible() )
            {
                for(vlocity_cmt__OrderItemRelationship__c qlir :
existingqprel)
                {

if(qlir.vlocity_cmt__OrderItemId__r.vlocity_cmt__Product2Id__c !=
null)
                    {

sourceproductid.add(qlir.vlocity_cmt__OrderItemId__r.vlocity_cmt__Pro
duct2Id__c);
                    }
                    if(qlir.vlocity_cmt__RelatedAssetReferenceId__c
!= null)
                    {

assetreflist.add(qlir.vlocity_cmt__RelatedAssetReferenceId__c );
                    }
                    if(qlir.vlocity_cmt__RelatedOrderItemId__c
!=null)
                    {

relorderidlist.add(qlir.vlocity_cmt__RelatedOrderItemId__c);
                    }
                    if(qlir.vlocity_cmt__OrderItemId__c !=null)
                    {

sourceorderidlist.add(qlir.vlocity_cmt__OrderItemId__c);
                    }
                }
            }
            Map<id,OrderItem > oisourcemap = new Map<id,OrderItem >(
[select
id,Pricing_Attributes__c,vlocity_cmt__AttributeSelectedValues__c,vloc
ity_cmt__Product2Id__c from OrderItem where (id in:relorderidlist or
id in:sourceorderidlist)]);

            Map<String,OrderItem > oimap = new Map<string,OrderItem
>();
            for(OrderItem oi : [select
id,vlocity_cmt__AttributeSelectedValues__c,Pricing_Attributes__c,vloc
ity_cmt__Product2Id__c,vlocity_cmt__AssetReferenceId__c from
OrderItem where vlocity_cmt__AssetReferenceId__c in:assetreflist])
            {
                oimap.put(oi.vlocity_cmt__AssetReferenceId__c,oi);
            }
            Map<string,Asset> assetmap = new Map<string,Asset>();
            for(asset a:[select vlocity_cmt__AssetReferenceId__c,id,
vlocity_cmt__AttributeSelectedValues__c,Pricing_Attributes__c,Product
2Id  from asset where vlocity_cmt__AssetReferenceId__c
in:assetreflist])
            {
                assetmap.put(a.vlocity_cmt__AssetReferenceId__c,a);
            }
```

```
if(vlocity_cmt__ProductRelationship__c.sObjectType.getDescribe().IsAc
cessible()){
                for (vlocity_cmt__ProductRelationship__c   prList :
[SELECT Id,Name,Product_Relationship_Type__r.Name,
vlocity_cmt__Product2Id__c,vlocity_cmt__AttributeModificationContext_
_c,

vlocity_cmt__RelatedProductId__c,vlocity_cmt__RelationshipType__c,vlo
city_cmt__ActionParameters__c ,

Product_Relationship_Type__r.vlocity_cmt__RelationshipTypeScope__c,Pr
oduct_Relationship_Type__c

FROM vlocity_cmt__ProductRelationship__c

WHERE vlocity_cmt__Product2Id__c in :sourceproductid])
                {


prreltypemap.put(prList.id,prList.Product_Relationship_Type__r.Name);
                    if(prList.vlocity_cmt__RelationshipType__c ==
'Attribute Propagation/Reference' &&
prList.vlocity_cmt__AttributeModificationContext__c =='ReliesOn' &&
prList.vlocity_cmt__ActionParameters__c!=null)
                    {
                        String s = prList.vlocity_cmt__Product2Id__c
+ ':' + prList.vlocity_cmt__RelatedProductId__c ;

prprodlist.add(prList.vlocity_cmt__Product2Id__c);
                        prrelmap.put(s,prList);
                    }


                }
            }

            List<id> qlilistup = new List<id>();
            Map<id,OrderItem> oiattrimap = new Map<id,OrderItem>();
            if(prprodlist.size()>0)
            {
                for(vlocity_cmt__OrderItemRelationship__c  qlir
:existingqprel)
                {
```

```
if(prprodlist.contains(qlir.vlocity_cmt__OrderItemId__r.vlocity_cmt__
Product2Id__c))
                    {
                        OrderItem qlsource =
oisourcemap.get(qlir.vlocity_cmt__OrderItemId__c);

if(qlir.vlocity_cmt__RelatedAssetReferenceId__c  != null)
                        {

if(oimap.containskey(qlir.vlocity_cmt__RelatedAssetReferenceId__c))
                            {
                                OrderItem qlrel =
oimap.get(qlir.vlocity_cmt__RelatedAssetReferenceId__c);


                                String s =
qlsource.vlocity_cmt__Product2Id__c + ':' +
qlrel.vlocity_cmt__Product2Id__c ;

                                if(prrelmap.containsKey(s))
                                {

vlocity_cmt__ProductRelationship__c Prrel = prrelmap.get(s);
                                    string sumcopy;
                                    string copyatt;
                                    string totalparam =
actionparamatersmanipulate(Prrel.vlocity_cmt__ActionParameters__c);
                                    copyatt =
totalparam.split(';')[0];
                                    sumcopy =
totalparam.split(';')[1];
                                    Integer k=0;

if(oiattrimap.containskey(qlsource.id))
                                    {
                                        k=1;
                                    }
                                    String a =
modifyAttributes(qlsource.vlocity_cmt__AttributeSelectedValues__c,qlr
el.vlocity_cmt__AttributeSelectedValues__c,copyatt,sumcopy,k);

                                    if(a!=null)
                                    {
```

```
qlsource.vlocity_cmt__AttributeSelectedValues__c = a;
                                    }

                                    qlilistup.add(qlsource.id);

oiattrimap.put(qlsource.id,qlsource);

                                }
                            }
                            else
if(assetmap.containsKey(qlir.vlocity_cmt__RelatedAssetReferenceId__c)
)
                            {
                                asset a =
assetmap.get(qlir.vlocity_cmt__RelatedAssetReferenceId__c);

                                String s =
qlsource.vlocity_cmt__Product2Id__c + ':' + a.Product2Id;
                                if(prrelmap.containsKey(s))
                                {

vlocity_cmt__ProductRelationship__c Prrel = prrelmap.get(s);
                                    string sumcopy;
                                    string copyatt;
                                    string totalparam =
actionparamatersmanipulate(Prrel.vlocity_cmt__ActionParameters__c);
                                    copyatt =
totalparam.split(';')[0];
                                    sumcopy =
totalparam.split(';')[1];
                                    Integer k=0;

if(oiattrimap.containskey(qlsource.id))
                                    {
                                        k=1;
                                    }
                                    String ax =
modifyAttributes(qlsource.vlocity_cmt__AttributeSelectedValues__c,a.v
locity_cmt__AttributeSelectedValues__c,copyatt,sumcopy,k);

                                    if(ax!=null)
                                    {

qlsource.vlocity_cmt__AttributeSelectedValues__c = ax;
```

```
                                    }
                                    qlilistup.add(qlsource.id);

oiattrimap.put(qlsource.id,qlsource);

                                }


                            }
                        }
                    }
                }
            }
            Map<String,Object> outputmap = new Map<String,Object>();
            Map<String,Object> theObj = new Map<String, Object>();
            theObj.put('Scope',Scope);
            theObj.put('Records',qlilistup);

            Map<String,Object> theObj1 = new Map<String, Object>();
            theObj1.put('message','Attributes are updated
successfully');
            List<Object>msglist =new List<Object>();
            msgList.add(theObj1);
            List<Object>recList =new List<Object>();
            recList.add(theObj);
            outputmap.put('totalSize', 1);
            outputmap.put('messages',msgList);
            outputmap.put('records',reclist);
            if(oiattrimap.size()>0)
            {
                Database.update(oiattrimap.values());

outMap.put(BT_GlobalConstants.CONST_CREATIONSTATUS,Json.serialize(out
putmap));
            }
        }


    }
    public string actionparamatersmanipulate(string s)
    {
        string s3;
        string sumcopy=null;
        string copyatt=null;
        string s4;
        if(s!=null)
        {
            String s1 = s.substringafter('COPY(');



            if(s1.contains(',SUM('))
            {
                copyatt = s1.substringbefore(',SUM(');
                s3= s.substringafter(',SUM(');
                sumcopy= s3.substringbefore(')');
            }
            else
            {
                copyatt = s1.substringbefore(')');
            }
        }
        return copyatt+';'+sumcopy;
    }
    public string modifyAttributes(String source,String target,String
copyparam,String sumcopyparam,Integer k)
    {
        Map<String, String> sourceMap = new Map<String, String>() ;
        Map<String, String> targetMap = new Map<String,String>();
        string jsonstring ;
        if(target !=null)
        {
            targetMap = (Map<String, String>)
JSON.deserialize(target,Map<String, String>.class);
        }
        if(source !=null)
        {
            sourceMap = (Map<String, String>)
JSON.deserialize(source,Map<String, String>.class);
        }
        try{
            if(copyparam!=null && target !=null && source !=null)
            {
                List<String> paramlist = copyparam.split(',');


                for(String s:paramlist)
                {
                    if(s.contains(':'))
                    {
                        integer n = s.indexof(':');
                        String souratt = s.substring(0,n);
                        String taratt = s.substring(n+1,s.length());
                        if(targetMap.containsKey(taratt))
                        {
                            if(sourceMap.containsKey(souratt))
                            {
                                String sourval =
sourceMap.get(souratt);
                                sourval = targetMap.get(taratt);
                                sourceMap.put(souratt,sourval);

                            }
                        }
                    }
                    else

                    {

                        if(targetMap.containsKey(s))
                        {

                            if(sourceMap.containsKey(s))
                            {
                                String sourval = targetMap.get(s);
                                sourceMap.put(s,sourval);

                            }

                        }
                    }
                }
            }
            if(sumcopyparam!=null && target !=null && source !=null)
            {
                List<String> paramlist = sumcopyparam.split(',');
                for(String s:paramlist)
                {
                    if(s.contains(':'))
                    {
                        integer n = s.indexof(':');
                        String souratt = s.substring(0,n);
                        String taratt = s.substring(n+1,s.length());
                        if(targetMap.containsKey(taratt))
                        {
                            if(sourceMap.containsKey(souratt))
                            {
                                String sourval =
sourceMap.get(souratt);
                                String tarval = targetMap.get(s);
                                if(k==0)
                                {
                                    sourval ='0';
                                }
                                decimal d;
                                if(tarval !=null && sourval!=null)
                                    d = Decimal.valueOf(sourval) +
Decimal.valueOf(tarval);
                                sourceMap.put(s,String.valueOf(d));

                            }
                        }
                    }
                    else{

                        if(targetMap.containsKey(s))
                        {

                            if(sourceMap.containsKey(s))
                            {
                                String tarval = targetMap.get(s);
                                String sourval = sourceMap.get(s);
                                if(k==0)
                                {
                                    sourval ='0';
                                }
                                decimal d;
                                if(tarval !=null && sourval!=null)
                                    d = Decimal.valueOf(sourval) +
Decimal.valueOf(tarval);
                                sourceMap.put(s,String.valueOf(d));

                            }
                        }
                    }
                }
            }

            if(sourceMap !=null)
            {
                jsonstring =  JSON.serialize(sourceMap);
            }


        }
        catch(Exception e){
            System.debug('The following exception has occurred: ' +
e.getMessage());

        }
        return jsonstring;

    }
}
```

Performance Metrics

Metrics	APEX Execution Time	APEX Execution Time
No of Line Items in Quote: 1 **No of Product Relationship Definition: 1** **Average no of Attributes per line item: 2** **Quote Product Relationships - 1**	670 ms	656 ms
No of Line Items in Quote: 100 **No of Product Relationship Definition: 3** **Average no of Attributes per line item: 5** **Quote Product Relationships - 52**	919 ms	923 ms
No of Line Items in Quote: 1000 **No of Product Relationship Definition: 3** **Average no of Attributes per line item: 5** **Quote Product Relationships - 800**	2953 ms	3015 ms

📒 **Note**
This feature is tested in 242 CMT versions and it is expected to work in higher versions of CMT packages as well.

Credits

Design & Author	Jeevanprakash Chinya Manjunatha
Solution Architect & Reviewer	Francisco Pina
Developers	Jeevanprakash Chinya Manjunatha Nilavra Guharay