1. **Add Auction Type Combo box in Market Replay Screen Selection**
   1. **IDL Changes**
      * **CMI IDL Modification**

None

* + - **Other IDL Modification**

**Order.idl(Change)**

Change this IDL to make new entries for buysell, buyselldesc and series in AuctionOrderDetailStruct to reuse it for FLEX Auction.

struct AuctionOrderDetailStruct {

……………………

……………………

string buySell;

string buySellDesc;

string series;

};

**AuctionService.idl(New)**

This IDL for FLEX Auctions only.

* Add new Method to get Auctions.

bo::Order::AuctionOrderDetailStructSequence getAuctions(

in string userId,

in string userRoles,

in bo::Common::SearchCriteriaStruct searchParam,

in string auctType

)

raises(

exceptions::SystemException,

exceptions::CommunicationException,

exceptions::DataValidationException,

exceptions::AuthorizationException,

exceptions::NotFoundException

);

* Add new Method forget Auctions Details.

bo::Order::AuctionOrderDetailStructSequence getAuctionDetails(

in string userId,

in string sessionId,

in string sbtAuctUid,

in string tradeDate

)

raises(

exceptions::SystemException,

exceptions::CommunicationException,

exceptions::DataValidationException,

exceptions::AuthorizationException,

exceptions::NotFoundException

);

* 1. **Classes & Interfaces**



### AuctionListAction.java

### This is a Struts Action implementation. It takes values from AuctionListForm bean & calls the AuctionModel & then forwards the request to ActionList jsp.

### MRParamHelper.java

This class takes values through getParameter and puts it into SearchCriteriaImpl.

### AuctionModel.java

This class initializes Home and calls service to fetch data for auctions and returns AuctionOrderDetailStruct array to Action class.

### AuctionServiceImlp.java

This class gets AuctionHome object and calls the findAuctionList method.

### AuctionHomeImpl.java

This class calls AuctionHelper through it’s getAuctionList method.

### AuctionListForm.java

This class is FormBean class of Struts framework which extend ValidateForm. It has getter and setter method respective to JSP forms.

### AuctionServiceImpl.java

Add new Business service to get FLEX auctions from database using Database service. This intializes AuctionsHome and calls its findAuctions method and returns AuctionOrderDetailStruct array to the Model class.

### AuctionService.java

This is service interface which extends AuctionServiceOperations class of IDL.

### AuctionServiceHome.java

This is the home Interface used to deploy this service in Market Replay server.

### AuctionServiceHomeImpl.java

This class is the implementation of AuctionServiceHome interface which creates the instance of AuctionServiceHomeImpl.

### AuctionsHome.java

This class is home Interface used to deploy Database service in Market Replay server.

### AuctionsHomeImpl.java

This class is implementation of AuctionHome class which extends DBDomain class. This class establishes database connection and executes database queries and return result to business service.

### AuctionOrderDetail.java

This is an interface which has few methods for Auctions

A Getter and Setter method for buysell

B Getter and Setter method for buyselldesc

C Getter and Setter method for series

### AuctionOrderDetailImpl.java

This Class is implementation of AuctionOrderDetail class which has Auctions details and uses to converts struct Object to Domain Object and Domain Object back to Struct.

### AuctionServiceProxy.java

This is Proxyservice class which implements Bussiness AuctionService interface which is used to call remote service

### AuctionServiceHomeProxyImpl.java

This class Implements AuctionServiceHome class and extends proxyHomebase. It intializes

AuctionService Object and Interceptor Object.

### MROptionsScreenSeclection.jsp

Update this file to add new Auction CheckBox and AuctionType ComboBox filters

### initialMRParms.js

Update this javaScript to Hide/Show AuctionType Combox according to selected Session.

Add a function which creates the URL with appropriate Struts action

### InputParmsInclude.jsp

Update this Jsp to display selected filter criteria into header of generated Report

### AuctionList.jsp

Create new JSP to display FLEX Auctions for selected filter criteria.

### OptionAuctionList.jsp

Create a JSP to display Auctions Data using pagination.

### AuctionServiceInterceptor.java

This is Interceptor File for Auction generated automatically by making entry in Inteceptor/makefile.

### MRACL.csv

Give rights to getAuctions method for MRFlex infraRole.

### MRCollectionPdfView

This class contains header related information of Pdf file.

Hide P/S related information for FLEX\_MAIN session.

### AuctionListPdfView.java

This is Formatter class for PDF generation .This class generate PDF file of Auction LiSt information.

* 1. **SCRIPTS**
     + **startMarketReplayServerWorker**

Make entry to initialize GlobalAuctionHome defined in Auctions.xml.

-DSMA.GlobalAuctionsHome=Auctions:GlobalAuctionsHome\_0 \

Make entry for AuctionServiceIORFile.ior generation

-Dboserver.AuctionServiceIORRefURL=file://${RUN\_DIR}/ior/AuctionServiceIORFile.ior\

* + - **StrarMarketReplayProxyServerWorker**

Make entry for AuctionServiceIORFile.ior generation

-Dboserver.AuctionServiceIORRefURL=file://${RUN\_DIR}/ior/AuctionServiceIORFile.ior\

* 1. **Database Changes**

Insert Auction Type into Regs\_Code table.

* 1. **SetContext**

None

* 1. **XML/DTD**
     + **struts-config.xml**

Add new entry for auctionlist action.

<action path="/auctionlist" type="com.cboe.mrclient.action.AuctionListAction"

name="AuctionListForm"

scope="request"

validate="false"

parameter="reqCode"

input="/OrderList.jsp">

<forward name="auctionlist" path="/auctionList.jsp"/>

</action>

* + - **MarketReplayServer.xml**

Make AuctionHome and AuctionServiceHome entry.

<!-- For Auction Filteration for CFLEX -->

<BOHome name=*"AuctionsHome"*>

<BOHomeProperties>

<homeImpl>com.cboe.bo.domain.impls.AuctionsHomeImpl</homeImpl>

<managedResourceName>AuctionsHomeImpl</managedResourceName>

</BOHomeProperties>

</BOHome>

<BOHome name=*"AuctionServiceHome"*>

<BOHomeProperties>

<homeImpl>com.cboe.bo.businessServices.impls.AuctionServiceHomeImpl

</homeImpl>

<poaName>AuctionServicePOA</poaName>

<objectId>AuctionService</objectId>

</BOHomeProperties>

</BOHome>

<listOfHomes>

AuctionsHome

AuctionServiceHome

</listOfHomes>

* + - **MRCatlog.xml**

Make Auctions.Xml Entry which is newly created for Auction.

<Map PublicID="Auctions" HRef="properties/xml/Auctions.xml"/>

* + - **MarketReplayProxyServer.xml**

Make Entry For ActionServiceHome for Proxy call

<BOHome name=*"AuctionServiceHome"*>

<BOHomeProperties>

<homeImpl>com.cboe.bo.businessServices.proxies.

AuctionServiceHomeProxyImpl

</homeImpl>

<interceptorName>AuctionServiceInterceptor</interceptorName>

<interceptorImpl>com.cboe.bo.businessServices.interceptors.

AuctionServiceInterceptor

</interceptorImpl>

</BOHomeProperties>

</BOHome>

<listOfHomes>

AuctionServiceHome

</listOfHomes>

* + - **OrbService.xml**

Create new entry for AuctionServicePOA

<OrbServicePOAImplProperties>

<poaList>...,UserFirmHistServicePOA,CoatsTradesServicePOA,

GraphDataServ icePOA,AuctionServicePOA</poaList>

<poolSize>5</poolSize>

</OrbServicePOAImplProperties>

<POA name=*"AuctionServicePOA"*>

<POAProperties>

<persistentIor value=*"true"*/>

<bindOrderType value=*"SWUL"*/>

<poolSize>25</poolSize>

<parentPoa>UserPOA</parentPoa>

</POAProperties>

</POA>

* + - **Auctions.xml**

Create new XML for Auctions .

* + - **Auctions.dtd**

Create new DTD for Auctions .

<OrbServicePOAImplProperties>

<poaList>...,UserFirmHistServicePOA,CoatsTradesServicePOA,

GraphDataServ icePOA,AuctionServicePOA</poaList>

<poolSize>5</poolSize>

</OrbServicePOAImplProperties>

* 1. **TXT File**
     + **MRServerxml.txt**

Make Entry for Auctions.xml

/vobs/cboe/BODomainV2/release/xml/Auctions.xml

* 1. **Make File Entry**
     + **BOServer\_idl.mk**

Make entry for Auction Service to build AuctionService.idl

AuctionService.idn \

* + - **interceptors/Makefile**

Make entry to create AuctionServiceInterceptor.java

com.cboe.bo.businessServices.interceptors.AuctionServiceInterceptor-com.cboe.bo.businessServices.interfaces.AuctionService \

1. **Add %Product Type Check box** 
   1. **Classes/Interfaces**

### InitialMRParms.jsp

Add new check box for %Product .

<td nowrap><div id="percentProductCheckBoxDiv">

<label for="percentProductInd">% Product</label>

</div>

</td>

<td nowrap><div id="percentProductLabelDiv">

<html:checkbox styleId="percentProductInd" property="percentProductInd" value="true"

onclick="MakePercentProductVisible(this.checked)"/>

</div>

</td>

### initialMRParms.js

Add MakePercentProductVisible function to get selected parameters and call action.

function MakePercentProductVisible(serieschecked)

{

if (serieschecked)

{

document.getElementById('seriesRetrieve').value=null;

document.getElementById("seriesEmptyDiv").style.display = '';

document.getElementById("seriesDiv").style.display = 'none';

}

else

{

if(ValidateDate(document.forms['InitialMRParmsForm'].tradeDate) == false){

return false;

}

document.getElementById("seriesEmptyDiv").style.display = 'none';

document.getElementById("seriesDiv").style.display = '';

var parms;

var i = document.getElementById('sessionName').selectedIndex;

parms = "classKey="+ document.getElementById('classKey').value

+ "&PercentProductOnly=Y"

+ "&underSecKey=" +

document.getElementById('underSecKey').value

+ "&tradeDate=" + getTradeDate()

+ "&sessionName=" +

document.getElementById('sessionName').value

+ "&sessionNameDesc=" +

document.getElementById('sessionName').options[i].value;

var url='initialmrparms.do';

var displayDivName = "seriesDiv";

makeRequest(url,parms,displayDivName);

//make PercentProduct unchecked.

document.getElementById('percentProductInd').checked=false;

}

}

Hide %Product checkbox for other sessions.

function hideDivesforStock(){

………….

………….

updateDivDisplay('percentProductCheckBoxDiv',false);

updateDivDisplay('percentProductLabelDiv',false);

………..

…………

}

function hideDivesforFutures(){

………….

………….

updateDivDisplay('percentProductCheckBoxDiv',false);

updateDivDisplay('percentProductLabelDiv',false);

………..

…………

}

function displayDivesforOptions (){

………….

………….

updateDivDisplay('percentProductCheckBoxDiv',false);

updateDivDisplay('percentProductLabelDiv',false);

………..

…………

}

Make %Product checkbox Visible for Flex Session.

Function displayDivesforCFlexOptions(){

………….

………….

updateDivDisplay('percentProductCheckBoxDiv',true);

updateDivDisplay('percentProductLabelDiv',true);

………..

…………

}

### InitialMRParmsAction.java

Add new case for %product same as AllSeries and call model class getPercentProducts method from it.

if ("Y".equalsIgnoreCase(request.getParameter("percentProductInd")))

{

products = prModel.getPercentProducts(

(String) session.getAttribute(ClientConstants.USER\_NAME),tradeDate,sessionName,classes);

}

### ProductModel.java

Add new getPercentProducts method for %product same as getProducts and call getPercentProducts of ProductServiceImpl.java from it

Public HashMap getPercentProducts(String userId,

String tradeDate,

String sessionType,

Collection classes

)

throws SystemException, CommunicationException, DataValidationException,

AuthorizationException, NotFoundException

{ ……………..

………………

productKeyNameStruct = productImpl.getPercentProducts(userId,tradeDate,sessionType,

rcStruct);

……………….

……………….

// set pairedProducts hashmap to override values for FLEX\_MAIN irrespective of call and put

pairedProducts.put(productKeyNameDomain.getExerPrice()

+"," +

productKeyNameDomain.getExprDate()

+ "," + productKeyNameDomain.getProdName().getReportingClass(),

productKeyNameDomain);

}

if (pairedProducts != null){

colPairedProducts = pairedProducts.values();

}

splitProducts.put("pairedProducts",colPairedProducts);

### ProductServiceImpl.java

Add new getPercentProducts method for %product same as getProducts and call findPercentProducts of ProductHomeImpl.java from it

public ProductKeyNameStruct[] getPercentProducts(

String userId,

String tradeDate,

String sessionType,

ReportingClassStruct[] classes)

throws SystemException, CommunicationException, DataValidationException, AuthorizationException, NotFoundException

{

------------

------------

colProducts=getProductHome().findPercentProducts(tradeDate,colReportingClasses,sessionType);

------------

------------

}

### ProductHomeImpl.java

Add new Method findPercentProducts same as findProducts and add new where clause which check ‘%’

Symbol in the field and depend upon that fetch only %Product from database.

public Collection<ProdKeyName> findPercentProducts(String pTradeDate,

ArrayList<ReportingClass> reportingClasses

,String pSessionType)

throws SystemException {

------------

------------

String whereClause = PERCENTPRODUCT\_WHERE\_CLAUSE;

------------

------------

}

Where clause to fetch only percentage product from database

private static String PERCENTPRODUCT\_WHERE\_CLAUSE =

" Where product.rpt\_class\_uid = rpt\_class.rpt\_class\_uid "

+" AND product.prod\_type\_code = rpt\_class.prod\_type\_code "

// + " AND product.prod\_type\_code = ? "

+ " AND product.list\_stat\_code = 1 "

+ " AND rpt\_class.list\_stat\_code = 1 "

+ " AND rpt\_class.list\_stat\_code = 1 "

+ " AND Raw\_Exrce\_Price LIKE '%\\%%' ESCAPE '\' ";

### ProductHome.java

Add new method findPercentProducts same as findProducts.

public Collection<ProdKeyName> findPercentProducts(String pTradeDate, ArrayList<ReportingClass> reportingClasses,String pSessionType) throws SystemException;

### ProductServiceProxy.java

Add new method getPercentProducts same as getProducts for proxy usage

public ProductKeyNameStruct[] getPercentProducts(String userId,

String tradeDate,

String sessionType,

ReportingClassStruct[] classes)

throws SystemException, CommunicationException, DataValidationException, AuthorizationException, NotFoundException {

return this.delegate.getProducts(userId,tradeDate,sessionType,classes);

}

### InitialMRParmsForm.java

Add new getter and setter method for percentProductInd checkbox of IntialMRParam.jsp

boolean percentProductInd;

public boolean isPercentProductInd() {

return percentProductInd;

}

public void setPercentProductInd(boolean percentProductInd) {

this.percentProductInd = percentProductInd;

}

### Product.Jsp

### Set Display table for FLEX\_MAIN takes values from colPairedProducts of InitialMRParmsForm

<c:if test="${(InitialMRParmsForm.sessionName == 'FLEX\_MAIN')}">

<display:table id="seriesRow" cellspacing="1" name="requestScope.InitialMRParmsForm.colPairedProducts" export="false" class="dataRows" requestURI="initialmrparms.do" decorator="com.cboe.mrclient.forms.InitialMRParmsWrapper">

<display:setProperty name="basic.empty.showtable" value="true"/>

<display:setProperty name="basic.msg.empty\_list" value="<div class=\"noDataFound\">

No data was found matching the current criteria.</span>"/>

<c:if test="${(!empty InitialMRParmsForm.colPairedProducts)}">

<display:column title="Series Class" class="widerColumn" property="series" style="text- align: left;" />

<display:column title="Call Exer. Price" property="flexCallCheck" style="text-align: left;" />

<display:column title="Put Exer. Price" property="flexPutCheck" style="text-align: left;" />

</c:if>

</display:table>

</c:if>

### InitialMRParmsWrapper.java

Add new methods for getFlexPutCheck and getFlexCallCheck for FLEX\_MAIN .

Update startRow method for FLEX\_MAIN.

### MRACL.csv

Give rights to getPercentProducts method for MRFlex infraRole.

* 1. **XML Changes :**

NA

* 1. **IDL Changes :**

1. **ProductService.idl**

Update idl add new getPercentProducts same as getProducts method.

bo::Product::ProductKeyNameStructSequence getPercentProducts(

in string userId,

in string tradeDate,

in string sessionId,

in cmiProduct::ReportingClassStructSequence classes

) raises(

exceptions::SystemException,

exceptions::CommunicationException,

exceptions::DataValidationException,

exceptions::AuthorizationException,

exceptions::NotFoundException

);

1. **Fetch All strategies(Fixed Product and Percentage product) when strategy checkbox is clicked .**
   1. **Sequence Diagram**

###### NA

* 1. **Files Changes**

1. **ProductHomeImpl.java**

Update findStrategies method use getStrategiesForOptions method for FLEX\_MAIN also.

if ("W\_MAIN".equalsIgnoreCase(pSessionType)|| "FLEX\_MAIN".equalsIgnoreCase(pSessionType))

{

productList = getStrategiesForOptions(pTradeDate, reportingClasses,

pSessionType, connName, conn);

}

* 1. **XML Changes**

NA

* 1. **IDL Changes**

NA

1. **Create new screen to display FLEX Auction list**
   1. **Sequence Diagram**

NA

* 1. **Files Changes**

1. **AuctionListWrapper.java**

This is decorator class.Add new method to create hyper link into Auction State column.

1. **AuctionDetailListAction.java**

### This is a Struts Action implementation. It takes values from AuctionDetailListForm bean & calls the OrderModel,AuctionModel and TradeModel & then forwards the request to ActionDeatilList jsp.

1. **AuctionDetailListForm.java**

This class is FormBean class of Struts framework which extend ValidateForm. It has getter and setter method respective to JSP forms.

### AuctionModel.java

This class initializes Home and calls services to fetch data for AuctionInitiator,AuctionResponses,Trade Details and stratergy.

### AuctionServiceImlp.java

This class gets AuctionHome,OrderHome,TradeHome object and calls the findOrder,

findAllAuctionResponses,findSingleTrade and

findOrderLegs methods.

### AuctionHomeImpl.java

This class is DataBase service.It hits Database fetch AuctionInitiator,AuctionResponse,

TradeDetails and Strategy information from it.

### AuctionOrderDetail.java

This is an interface which has few methods for Auctions

A Getter and Setter method for tradeSessionId

B Getter and Setter method for sbtOrderUid

### AuctionOrderDetailImpl.java

This Class is implementation of AuctionOrderDetail class which has Auctions details and uses to converts struct Object to Domain Object and Domain Object back to Struct.

### AuctionServiceProxy.java

This is Proxyservice class which implements Bussiness AuctionService interface which is used to call remote service.Add new method getAuctionDetails to fetch Aucton Details data.

1. **AuctionDetail.jsp**

This Screen shows AuctionDetails like AuctionInitiator , Auction Responses, Trade Details

And Strategy details. This jsp include other jsps.

1. **AuctionIntiator.jsp**

This Jsp shows Auction Initiator Information.

1. **AuctionResponses.jsp**

This Jsp shows Auction Responses comes for particular options.

1. **Auction.js**

This js contains function which open Auction Detail jsp.

* 1. **SCRIPTS**

NA

* 1. **Database Changes**

NA

* 1. **SetContext**

NA

* 1. **XML/DTD**
     + **struts-config.xml**

<action path=*"/auctiondetaillist"* type=*"com.cboe.mrclient.action.AuctionDetailListAction"*

name=*"auctionDetailListForm"*

scope=*"request"*

validate=*"false"*

parameter=*"reqCode"*

input=*"/AuctionList.jsp"*>

<forward name=*"auctiondetaillist"* path=*"/AuctionDetail.jsp"*/>

* + - **Auctions.xml**

Added new select Clause and from clause for Auction response for CDB and DWS

* 1. **IDL Changes**
     + **CMI IDL Modification**

None

* + - **Other IDL Modification**

**Order.idl(Change)**

Change this IDL to make new entries for below fields.

struct AuctionOrderDetailStruct {

……………………

……………………

string tradeSessId;

long long sbtOrderUid;

};

**AuctionService.idl(Change)**

Add new method to fetch Auction Details.

bo::Order::OrderSummaryStructSequence getAuctionDetails(

in string userId,

in string sessionId,

in string sbtAuctUid,

in string tradeDate

)

raises(

exceptions::SystemException,

exceptions::CommunicationException,

exceptions::DataValidationException,

exceptions::AuthorizationException,

exceptions::NotFoundException

);

1. **Change Order list and Order Detail screens to include new order columns**
   1. **Sequence Diagram**

NA

* 1. **Files Changes**

1 **OrderSummary.java**

This is interface contains signature of getter and setter methods

Added below fields.

HEDGE\_EXEC\_INSTR\_TEXT

HEDGE\_REF\_TYPE

HEDGE\_REF\_PRICE

PRFCLTN\_PRICE

DELTA\_VALUE\_AMT

1. **OrderSummaryImpl.java**

This class contains getter and setter methods implementation of Order details.

Added below fields.

HEDGE\_EXEC\_INSTR\_TEXT

HEDGE\_REF\_TYPE

HEDGE\_REF\_PRICE

PRFCLTN\_PRICE

DELTA\_VALUE\_AMT

1. **OrderPopulateHelper.java**

Update this class to set the new columns in the OrderSummaryStruct.

1. **FLEXOptionList.jsp**

Add new Jsp to show below fields for Flex session only.

HEDGE\_EXEC\_INSTR\_TEXT

HEDGE\_REF\_TYPE

HEDGE\_REF\_PRICE

PRFCLTN\_PRICE

DELTA\_VALUE\_AMT

* 1. **SCRIPTS**

NA

* 1. **Database Changes**

NA

* 1. **SetContext**

NA

* 1. **XML/DTD**
* **Order.xml**

Update select clause for order add hedge columns into it.Add new Where clause for FLEX session.

* 1. **IDL Changes**
     + **CMI IDL Modification**

None

* + - **Other IDL Modification**

**Order.idl(Change)**

Change this IDL to make new entries for below fields.

HEDGE\_EXEC\_INSTR\_TEXT

HEDGE\_REF\_TYPE

HEDGE\_REF\_PRICE

PRFCLTN\_PRICE

DELTA\_VALUE\_AMT

struct OrderSummaryStruct {

……………………

……………………

string hedgeExecInstrText; //For FLEX

string hedgeRefType; //For FLEX

string hedgeRefPrice; //For FLEX

string prfcltnPrice; //For FLEX

string deltaValueAmt; //For FLEX };

1. **Revise input selection criteria for orders to remove unnecessary criteria**
   1. **Sequence Diagram**

NA

* 1. **Files Changes**

1 **MROptionsScreenSeclection.jsp**

Change this Jsp and shows only those fields which are relevant to FLEX session.

* 1. **SCRIPTS**

NA

* 1. **Database Changes**

NA

* 1. **SetContext**

NA

* 1. **XML/DTD**

NA

* 1. **IDL Changes**

NA

1. **Change Order History and Auction detail screens to display percentage price for leg details.**
   1. **Sequence Diagram**

NA

* 1. **Files Changes**

1 **OrdersHomeImpl**.**java**

Update populateOrderLegs method which fetches data from result set and populate OrderLegs Object. Add Code to fetch exer\_price\_ind and raw\_exer\_price from resultset and set their values.

1. **OrderLegsImpl.java**

This class is implementation of OrderLegs.java interface.

Add getter and setter methods for

exerPriceInd

rawExerPrice

1. **OrderLegs.java**

Add two new fields for % sign.

exerPriceInd

rawExerPrice

1. **AuctionDetails.jsp**

Add new property="rawExerPrice" into display tag to show % sign.

1. **OrderHistory.jsp**

Update jsp put check into display tag set property="rawExerPrice" for FLEX\_MAIN session else set property="exerPrice".

* 1. **SCRIPTS**

NA

* 1. **Database Changes**

NA

* 1. **SetContext**

NA

* 1. **XML/DTD**

**Orders.xml**

Update OrderLegsclause to add product.exer\_price\_ind and product.raw\_exer\_price.

product.exer\_price\_ind is used to identify % product and product.raw\_exer\_price shows

% Price.

* 1. **IDL Changes**

**Order.idl**

Update OrderLegStruct to add Below fields for % sign.

string exerPriceInd;

string rawExerPrice;

1. **Change trade and trade detail screens to display percentage strike and percentage price.**
   1. **Sequence Diagram**

NA

* 1. **Files Changes**

1 **TradePopulateHelper**.**java**

Update populateTradeSummary method which fetches data from result set and populate TradeSummary Object. Add Code to fetch exer\_price\_ind and raw\_exer\_price from resultset.Pass raw\_exer\_price to fmtSeriesV2 if product is % trade else pass exer Price.

* 1. **SCRIPTS**

NA

* 1. **Database Changes**

NA

* 1. **SetContext**

NA

* 1. **XML/DTD**

**Trade.xml**

Update TradeSummaryandTradeSummaryv2 to add product.exer\_price\_ind and

product.raw\_exer\_price.

product.exer\_price\_ind is used to identify % product and product.raw\_exer\_price shows

% Price.

* 1. **IDL Changes**

NA

1. **Add new filter to input selection criteria for current trade status**
   1. **Sequence Diagram**

NA

* 1. **Files Changes**

1 **MROptionScreenSelection.jsp**

Add below options entry into existing Trade Type Combo and set their vales according

Database for FLEX\_MAIN session.

ALL Pending Trade

           Sent to OCC

Pending Leg Price

            Pending Leg Price Approval

            Pending Strike Price

            On Hold

2 **TradeHomeImpl.java**

Set values respective to trade status into tradeTypeMap hashmap,

this hashmap has where clause according to selected tradestatus.

Add where clause TRADE\_STAT\_CODE  = 1 to 6 according to selected Trade type

* 1. **SCRIPTS**

Insert different Trade status into Regs\_Code table.

* 1. **Database Changes**

NA

* 1. **SetContext**

NA

* 1. **XML/DTD**

NA

* 1. **IDL Changes**

NA

1. **Add new filter to input selection criteria to select trades with percentage products**
   1. **Sequence Diagram**

NA

* 1. **Files Changes**

1 **MROptionScreenSelection.jsp**

Add below options entry into existing Trade Type Combo and set their vales according

Database for FLEX\_MAIN session.

Percentage Trade

2 **TradeHomeImpl.java**

Set values into tradeTypeMap hashmap,

this hashmap has where clause according to selected tradestatus.

Add condition raw\_product\_ind=p if selected trade type is Percentage trade.

* 1. **SCRIPTS**

Insert different Trade status into Regs\_Code table.

* 1. **Database Changes**

NA

* 1. **SetContext**

NA

* 1. **XML/DTD**

NA

* 1. **IDL Changes**

**NA**

1. **Change trade and trade detail screens to display trade status**

**raw\_trade\_price, Trade\_price\_ind columns**

* 1. **Sequence Diagram**

**NA**

* 1. **Files Changes**

**1 TradePopulateHelper.java**

Update populateTradeSummary method which fetches data from result set and populate TradeSummary Object. Add Code to fetch ORIG\_TRADE\_PRICE\_IND ,RAW\_ORIG\_TRADE\_PRICE and TRADE\_STAT\_CODE from resultset and set into Tradesummary Object.

1. **TradeSummaryImpl.java**

This class is implementation of TradeSummary java interface.

Add getter and setter methods for

origTradePriceInd

rawOrigTradePrice

tradeStatCode

1. **TradeSummary.java**

This is interface.It contains trade related information

Add below new fields .

origTradePriceInd

rawOrigTradePrice

tradeStatCode

1. **OptionTradeSummary.jsp**

Add new columns Trade Status,Raw Trade Price and Trade Price Ind into display tag for FLEX\_MAIN session only.

* 1. **SCRIPTS**

Insert different Trade status into Regs\_Code table.

* 1. **Database Changes**

NA

* 1. **SetContext**

NA

* 1. **XML/DTD**

**Trade.xml**

Update TradeSummaryandTradeSummaryv2 clause to add RAW\_ORIG\_TRADE\_PRICE,TRADE\_STAT\_CODE and ORIG\_TRADE\_PRICE\_IND.

ORIG\_TRADE\_PRICE\_IND is used to identify % trade ,RAW\_ORIG\_TRADE\_PRICE shows

% Price. AND TRADE\_STAT\_CODE shows current status values.

* 1. **IDL Changes**

**Trade.idl**

Update TradeSummaryStruct to add Below fields.

String origTradePriceInd

String rawOrigTradePrice

String tradeStatCode

1. **Add new order history event type descriptions to support**

**complex order leg price updates.**

* 1. **Sequence Diagram**

NA

* 1. **Files Changes**

NA

* 1. **SCRIPTS**

Insert different Trade status into Regs\_Code table.

* 1. **Database Changes**

NA

* 1. **SetContext**

NA

* 1. **XML/DTD**

NA

* 1. **IDL Changes**

NA

1. **Add new filter to input selection criteria for historical trade**

**status.**

?

(Depend Upon Server Team)