CBOEDirect Release Notes

CBOEDIR_8.3 and Infra 13.1

February 6, 2010

Table of Contents

Overview of changes in this release

Work requests included in this release

6231 – PMM for Complex Orders

6421 – Qualified Contingent Trades

6436 – RTT Back Stop

6444 – Faster Startup

6307 – C2 Enhancements Phase 2

6459 – CBSX Contingency Cross Enhancements Phase 2

6435 – Greater NEMA Allocation

6352 – FIX Order Processing and Appia Latency Reduction

6378 – Client Latency Improvements

WR6231 – PMM For Complex Orders

Goal

The goal is to enhance the cboedirect platform to assign entitlements to preferred market makers based on the following rules. New inbound complex orders, as well as orders resting on COB that become marketable, that have a preferred designation (PMM acronym) assigned to them will be candidates for this new process. The entitlement received by the PMM should be determined based on the quantity remaining at each calculation point after any customer contracts have been satisfied. The option component series of cross product spreads are eligible for PMM entitlements the same as complex orders without a stock leg.

Summary of changes for server side

Functionality changes

a) For incomings that auto-ex against the legs, the PMM gets an entitlement on any leg(s) where the PMM is part of the BBO.

b) For incomings that auto-ex against the COB, the PMM gets an entitlement if at the best COB price, regardless of whether that price is better than, equal to, or worse than the legs.

c) If at the conclusion of a COA, the incoming trades against the legs, the PMM gets an entitlement on any leg(s) where the PMM was part of the BBO at the start of the auction. The PMM does not get an entitlement on any leg(s) where the PMM was NOT on the BBO at the start but then joined after the auction started.

d) If at the conclusion of a COA, the incoming trades against the COB, the PMM gets an entitlement if on top of the COB at the start of the auction, regardless of whether the COB was better than, equal to, or worse than the legs.

e) If at the conclusion of a COA, the incoming trades against responses, the PMM gets an entitlement if either:

    (1) the PMM was at the BBO in ANY leg at the start of the auction, regardless of whether the legs were better than, equal to, or worse than the COB; or

(2) Was top of COB regardless of whether it is better than, equal to, or worse than legs.

f) For future testing, if the affiliated PMM firm has multiple individual MMs, and if at least one of the MMs is at the BBO at the start of the auction, then the affiliated firm will receive the greater of the entitlement or CUMA. If multiple individual MMs that are part of the affiliated PMM firm respond at the best price, the firms allocated contracts will be divided among the affiliated PMM firm's MMs using UMA.

g) PMM entitlements will not be determined at the End of an Auction to see if a PMM entitlement is on the legs or at the top of the COB. Instead, a COA entitlement will be given if:

(1) The PMM is on the legs or top of the COB at the START of the auction.

The entitlement will not be capped based on the size the PMM is showing at the leg price or top of the COB

Turning on - PMM for complex orders :

This functionality will be turned on automatically at the time of BC install and PMM entitlements apply when the properties have been setup.

The following properties need to set at the time of GC install and setup appropriately.

- **PDPMComplexEligibilty** - New routing property is added to control PMM functionality for applying PMM for Complex Orders on a trading class.
  - o Allocation for complex orders can not be controlled from the allocation strategy configuration alone. For Spread to Leg Trade the existing option allocation strategy configuration is used. Turning of at this level will disable PMM for options.

- **Allocation Strategy (Trading Property) - STRATEGY AUCTION TRADE** (8) will be used to configure PMM Allocation strategies for Complex Orders.
  - New inbound complex orders, as well as orders resting on COB that become marketable, that have a preferred designation (PMM acronym) assigned to them will be candidates for this new process
  - For incomings that auto-ex against the legs, the PMM gets an entitlement on any leg(s) where the PMM is part of the BBO.
  - For incomings that auto-ex against the COB, the PMM gets an entitlement if at the best COB price, regardless of whether that price is better than, equal to, or worse than the legs

WR6421 – Qualified Contingent Trades

Goal

The goal of the 'Qualified Contingent Trade' project is to provide an efficient process for trading options tied to a separate stock transaction if they conform to the rules defined for the Qualified Contingency Trade exemption. The objective is to allow hedged order pairs to cross immediately, ahead of any resting customer orders for QCT orders. The business goal of this project is to keep CBOE competitive and generate more business by allowing QCT orders. Currently the system does not process QCT orders.

Summary of changes for server side

a) Orders with "A:AIQ" in the 'Optional Data' field is an indication for the Order to be treated as a QCT Order.

b) QCT Orders will be allowed for all origin types which are allowed for AIMs

c) QCT orders are paired orders with “A:AIQ” marking on the primary order sent as an internalized order pair. There is no indication on the QCT match indicating it is a QCT match except for the Primary Order Info in the optional data.

d) QCT primary order can be sent in as independent order with A:AIQ marking. Match Order with the primary order info can be received later.

  a. Independent Primary Orders without a Match after a wait time will be cancelled. Match Orders with invalid primary order information will be cancelled.

e) QCT order size should equal to or be greater than QCC Minimum Volume, QCT orders with volume less than the QCC Minimum volume will be cancelled.

f) QCT orders allowed to cross ahead of resting customer orders.

g) Immediate cross of QCT orders is permitted as long as the crossing price is at or inside the NBBO for each option leg.

h) QCT orders will be cancelled if the crossing price is outside the Option NBBO.

i) QCT orders must be traded at standard increment prices.

j) Cross Product spreads (A:AIM) are not eligible for QCT funcitonality.

k) QCT functionality enablement is per class based on the QCC Minimun Value, the minmum value should be greater than ZERO to be eligible for QCC Cross trades.

l) QCT orders cannot be routed to PAR.

m) All direct routes, Cancel requests and Cancel-Replace requests for QCC Primary orders will be rejected.

n) QCT orders with any contingencies will be rejected.

Turning on – Qualified Contingent Trades :

The following properties need to set at the time of GC install and setup appropriately.

QCC Minimum Volume - New trading property is added to control minimum volume required for QCT orders to execute. The user will be able to configure ‘QCC Minimum Volume’ to any required value for a trading class. This is applicable for both simple and complex QCT orders. By default on rollout of this project ‘QCC Minimum Volume’ needs to be set to ‘0’ or ‘-ve’ value for all the class in order to disable this feature. The user can set it to required +ve value to enable ‘QCT order’ feature for desired class

WR6436 –RTT Back Stop

Goal

The goal of this WR is help the users when system reponse time is slow.

Summary of changes for server side

Quotes:

1. Cancel(backstop) user’s quotes for the class on the trade server if quote/quotes is received such that local clock is ‘n’ millis (backstop trigger timer) more than originating TTE timestamp, where ‘n’ is an XML-configured value (for the BC cluster, not fine-grained by class, firm or users).

2. The incoming call on quote/quotes will get NotAcceptedException when backstop is triggered with error code NotAcceptedCodes.QUOTE_CONTROL_ID

3. Cancel reports will be published with cancel reason ‘SYSTEM’ on cancelled quotes.

4. The initial value of the backstop trigger timer is set to be 3 seconds and can be tuned later on.

5. I orders will NOT be cancelled when backstop is triggered on quote/quotes.

6. AR command will be added to be able to disable this feature runtime

Orders:

1. Cancel(backstop) users I (quote like) orders for the class on the trade server if I order is received such that local clock is ‘n’ millis (backstop trigger timer) more than originating TTE timestamp. This is not the same setting like we have for quote in order to give us flexibility to turn the parameter for orders and quotes separately.

2. The incoming call on I order will get NotAcceptedException when backstop is triggered with error code NotAcceptedCodes.SERVER_NOT_AVAILABLE.

3. Cancel reports will be published with cancel reason ‘SYSTEM’ on cancelled I orders.

4. MKT order from CMI as the previous location that triggered backstop will be rejected upfront with NotAcceptedException error code NotAcceptedCodes.SERVER_NOT_AVAILABLE.

5. Cancel/Replace order with MKT new order when backstop is triggered, the cancelling order will be cancelled if order is at CBOEDirect, and MKT replacement order will be rejected.

6. Quotes will NOT be cancelled when backstop is triggered on I Order or MKT order.

7. AR command will be added to be able to disable this feature runtime.

8. Server IDL changes to support passing of Previous Order Location from OHS to Tradeeng is needed so we will be able to differentiate the order is coming from CMI or PAR, etc.

WR6444 –Faster Startup

This change has been disabled for this load as it exposed a race condition in another subsection of code.

Goal

The goal of this project is to speed up CBOE Direct trade engine processes by staring some of the processes in parallel.

Summary of changes for server side

Startup script changes

CBOE Dir startup script (startTradeengSystem) modified to include option (-Q) to start process in parallel.
When –Q option used for startup, only selected process will be started in a quick mode.
GC processes:V20Frontend,CoppGlobal,Membership Adapter,CFN Adapter, MDB Adapter and Market Data Report Server.
BC Processes:Status Server, Copp Bridge, Tips Stock Adapter and Stock CFN Adapter

By default the script will behaved as normal startup.

restartDevSystemWorker, this cript is modidied to start CBOEDir process for all test environment in a fast mode (parallel startup).

Start Session Changes

Currently when sessions are started we update session element which results in increase time to start session. We are using quick start option to start session for all session by default which bypasses updating the session elements.

Test Setup

1. **In setContext.main** file has following changes

- Introduced **-d** argument (***startSessionQuickFlag***) for Global server
- By default startSessionQuickFlag is true (i.e it will start session with quick flag ON)

2. When session are started by cron job every day Check by running following script to see that no classes are in NO_SESSION state , all of them must be in CLOSED state.

   $RUN_DIR/***showProductStatesBySession***

Create Business Day Changes.

Currently business day creation is done in sequential for each session and session template which take between 20 to 25 minutes to complete. In this realease we make changes to start business day creation in parallel for each session and session template.

A thread will be started to process business day for each session, and within the session another thread will be started to process each session template. We use thread pool to limit number of thread started during business day creation. We have limitied the number of concurrent thread for trading sessions as well as for trading session templates.

The following changes is introduces in the script, xml and dtd files.

1. In setContext.main –**DparallelCreateBusDayFlag** argument is added to allow us to create business day in parallel or in regular way.
2. a new home ThreadingSessionThreadPoolHome,tradingSessionThreadPoolHome.xml and tradingSessiionThreadPoolHome.dtd is created to configure number of thread we start for session and session template.

Content of the TradingSessionThreadPoolHome.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE GlobalTradingSessionThreadPoolHome SYSTEM
"../dtd/TradingSessionThreadPoolHome.dtd">
<GlobalTradingSessionThreadPoolHome>
   <TradingSessionThreadPoolHomeImpl
name="TradingSessionServiceThreadPoolHomeImpl">
      <TradingSessionServiceThreadPoolHomeImplProperties>
       <sessionCorePoolSize>15</sessionCorePoolSize>
         <sessionMaximumPoolSize>50</sessionMaximumPoolSize>
         <sessionKeepAliveTime>10000</sessionKeepAliveTime>
       <sessionTemplateCorePoolSize>30</sessionTemplateCorePoolSize>
         <sessionTemplateMaximumPoolSize>50</sessionTemplateMaximumPoolSize>
         <sessionTemplateKeepAliveTime>10000</sessionTemplateKeepAliveTime>
      </TradingSessionServiceThreadPoolHomeImplProperties>
   </TradingSessionThreadPoolHomeImpl>
</GlobalTradingSessionThreadPoolHome>
```

Verification:After system starup please check

1. **Business day created properly.**
2. **Session started successfully.**
3. **No product in no session state.**

WR6307 – C2 Enhancements Phase 2

Goal

The goal of this project is to fix problems related to AIM ISO and AIM SWEEP orders that were not completed under 8.2. Additional modifications are also being made to New HAL for BOB to pick up requirements that were missed in previous releases of this functionality.

Summary of changes for server side

The PO portion of an AIM order must always trade all of the quantity and should never be partially traded. There are certain circumstances prior to these changes that could result in a PO order not being fully traded.

For an AIM sweep order, the MO order must not be immediately cancelled at the end of the auction. The MO order can only be cancelled after the PO is fully traded for the sweep and linked away quantities as well as the auctioned quantity. Prior to this change, there were situations in which the MO would get cancelled before all PO available quantity was traded. For instance, if we linked away a portion of the PO quantity and subsequently the linked away quantity is rejected or not responded to, the MO would have been cancelled if the auction ended. This would leave PO quantity unaccounted for and no MO order to trade against.

For New HAL for BOB, the HAL auction needed to end prematurely if a manual quote came in (during the auction) that could trade against the auctioning order. If the properly priced manual quote comes in during the auction, the auction would be ended and the order will be routed to the Firm-Class PAR.

WR6459 – CBSX Contingency Cross Enhancements Phase 2

Goal

The goal of this project to add 4 new cross contingency types into the system and to add sub-penny crossing restrictions to the system where needed.

Summary of changes for server side

The 3 new contingency cross types being added already are being used at other exchanges, but will be new for CBSX. They are not subject to the Reg NMS protections. They should be allowed to trade at a price of up to 4 decimal places. They may or may not be sent in as a paired order list (it is not necessary that the same user sends in both sides of the cross). The trade ID can be used to tie the two sides together if necessary. The 3 new types to be added to the system are:

Next-day Cross.

Regardless of the NBBO or CBSX Book, these should execute at the limit price of the orders. They should be submitted to the DTCC for Next-Day settlement and be reported to the tape with the Next-day modifier.

- The existing 'trade type' of "N" (NEXT_DAY_TRADE) should be used on these trades.

Two-day Cross.

Regardless of the NBBO or CBSX Book, these should execute at the limit price of the orders. They should be submitted to the DTCC for Two-Day settlement and be reported to the tape with the Next-day modifier.

- A new unique 'trade type' should be created for the Two-day cross (don't use 'Regular' = "R")

Cash Cross.

Regardless of the NBBO or CBSX Book, these should execute at the limit price of the orders. They should be submitted to the DTCC for immediate settlement and be reported to the tape with the Cash modifier.

- The existing 'trade type' of "C" (CASH_TRADE) should be used on these trades.

WR6435–Greater NEMA Allocation

Goal

The goal is to give the greatest allocation for PMM or DPM to restrict professional customer orders.

Summary of changes for server side

1. Regular allocation is determined based on the calculation with the greatest allocation to PMM/DPM between configured Regular and NEMA allocation.
2. Supported allocation strategies are configured at OrderBook.xml under <greaterNEMAAllocationAlgorithm> tag.
3. Each allocation is calculated at least twice if the property allocation type between regular and NEMA can not be predetermined. The first calculation is for Regular allocation, the second one is NEMA allocation. If NEMA gives greater PMM/DPM allocation, NEMA allocation is used. Otherwise, the Regular allocation is recalculated in order to use it.
4. With multiple allocation attempts, performance impact is anticipated.
5. The greater NEMA does not apply to Auction trade, PAR trade, and opening trade.
6. The greater NEMA also applies to five lots or less allocation.

WR6352– Fix Order Entry and Appia Latency Reductions

Goal

The goal is to give the reduce the latency on Orders and to use NYFIX’s new Appia engine.

Summary of changes

1. New version 5.1 of Appia FIX engine.
2. Re-write of entire Order Interface. Order Entry (Simple and Complex), Cancels and Cancel/Replaces will be now concurrent. **This would mean that there is a possibility that of the cancel getting rejected if the firm puts in a cancel immediately following a order without waiting for ack on the original order. The firm must be able to handle this scenario and should retry the cancel after receiving the ack on the order.**
3. A new engine mode called STOCK is being introduced for W_STOCK session.
4. When the engine is in STOCK mode we will load all the stock and index products when a connection is released.
5. Session Disconnects are now handled on a separate thread.
6. Cancels that are currently getting rejected with NotAccepted Exception during BC failover are currently getting rejected with CxlRejReason tag 102=1 (Unknown order). This is the same error code when the order is not found in our cache. In order to distinguish Not Accepted from Not Found we are adding a new error code for tag 102 on Not Accepted Exception. Tag 102 = 5 (Cancel requests are currently not being accepted. Please retry or contact CBOE Helpdesk) on these Not Accepted Exceptions.

Eg.

BeginString[8] = FIX.4.2

BodyLength[9] = 0177

MsgType[35] = 9 Order Cancel Reject

MsgSeqNum[34] = 5

SenderCompID[49] = DFIX201

TargetCompID[56] = TEST209

SendingTime[52] = 20091002-15:46:34

OrderID[37] = NONE

ClOrdID[11] = IEC2479-20091002

CxlRejReason[102] = 5

Text[58] = IDL:exceptions/NotAcceptedException:1.0 on OrderCancel

OrigClOrdID[41] = EIC2479-20091002

OrdStatus[39] = 8 Rejected

CxlRejResponseTo[434] = 1 Order Cancel Request

CheckSum[10] = 087

7. Logging the times in Nanos. All the times will be now in nanos in the cas.log.
8. New Order and Quote TTE points in the FIX CAS. New order tte points for product query and getOrderByID has been introduced in this release.

WR6378 – Client Latency Improvements

Goal

The goal is to improve latency of the client tier by doing some enhancements to the existing features.

Summary of changes

1. Federated query optimizations

   The following federated calls were optimized. Now, they will be routed to only the BCs that correspond to the sessions the user is enabled for.
   - publishOrdersForUser
   - getOrderById
   - getOrderByIdV2
   - getOrderByORSID

   The above optimizations are in place only in the case of CASes running as ‘Local’. The above calls from ‘remote’ CASes will continue to go to all the BCs as the federation is done only at the FE layer.

2. Moving to the new logging service

   CAS and FIXCAS were changed to use the new logging service.

3. Suppressing the exception logging for NotAcceptedExceptions

The intereceptor template was modified so that CAS and FIXCAS interceptors no longer log the exception stack for the NotAcceptedExceptions.

WR6455 – Linkage Changes for Wolverine

Goal

The goal is to allow linking of orders away through a Wolverine Brokerage connection.

Summary of changes

For Business Cluster (BC)

For Order History Event:
Order History of Underlying/Original Order "Route Description field" will be populated correctly with 'Away Exchange Order ID' and 'Vendor Order ID' for below events:

1. Execution Report on Linked Order
2. Execution

For Trade Report:
Underlying Order Trade Report for CTM Clearing will have following fields populated correctly for Wolverine router vendor (assuming Underlying/Original Order was a 'buy'):

1. If the Underlying Order Origin Type = Market Maker then,
   a. SellerAccount = Linked Order Fill Report Account field (e.g. Q****)
   b. SellerOptionalData = Linked Order Fill Report Optional Data field (e.g. Q****)
   c. SellerSubAccount = Linked Order Fill Report Subaccount field (e.g. empty)

2. If the Underlying Order Origin Type = Non-Market Maker then,
   a. SellerAccount = Linked Order Fill Report Account field (e.g. empty)
   b. SellerOptionalData = Linked Order Fill Report Optional Data field (e.g. 44YM0409)
   c. SellerSubAccount = Linked Order Fill Report Subaccount field (e.g. 44YM0409)

Bug fixes included in this release

SEDL	PITS	Test Plan	Assigned	Date	Origin	Status	Description
8585	None	No	Gijo	2/12	Production	Submitted	Enhanced the total number of unique transactionIDs per CAS/FIX to over 16 million.
8400	None	Yes	Connie	11/30	Development	Submitted	C2 bug only - BestOfTheRestConsumerImpl where CBOE and CBOE2 are not unique
8419	None	no	Gijo	11/2	Production	Submitted	CAS interceptor instrumentors were found to be generating 0 values.
8470	None	No	Kai		Performance	In Progress	Bad ACL causes null pointers in TTE data.
8066	14420	Yes	Raghu		Production	Submitted	OMT PAR Alternate destination
N/A	14852	No	Rajesh		Assurance	Submitted	Trade cps order at multiple price points. Test Plan : Existing Cross Product Reserve Test Plan
8277	15070	Yes	Kai		Production	Submitted	New Routing for OMT
	15429	Yes	Hinjosa		Production	Submitted	Validation added to update sale prefix to F instead of SWEP when trade comes from W_STOCK session
8483	15348	No	Beniwal		Production	Submitted	Time is now properly formatted in UTC.
8503	15306	Yes	Diddiu	11/08		Submitted	Reject Complex Cust-Orign type restricted orders to book in all product states.
8399	15452	Yes	Connie	11/24	Production	Submitted	Because of Z badge update related to a user acr that has 30 user id, CD Server published multiple events for the same user id and thus caused network congestion
8424	15576	Yes	Ojeda		Assurance	Submitted	Incorrect exchange on trade.
8460	15604	Yes	Lin Chen		Production	Submitted	Increased HTS to ETS connection pool.
8436	15667	Yes	Sudhakar		Assurance	Submitted	Fixed sweep and AIM PO.
8444	15740	Yes	Crystal		Assurance	Submitted	Recreated indexes off partitions
	15819	No	Anil		Assurance	Submitted	Corrected strategy creations issue with more than one session.
8458	15857	No	Infra		Production	Submitted	Infra 13.1
8476	15889	Yes	Connie		Assurance	Submitted	Prod issue too. Allowed W_Main crossing order entry.
None	15978		Sudhakar		Assurance	Submitted	Another AIM FIX
8468	15997	No	Magee		Production	Submitted	Removed confusing CFIX error message.
8501	16033	Yes	Sudhakar		Assurance	Submitted	ISB linkage order test plan (3.2.2-3.2.4)
N/A	16099	Yes	Diddi		Production	Submitted	Replaced hardcoded strategy with configurable setting. Corrected rollback.
8508	16150	Yes	Nyoman		Assurance	Sumitted	Corrected random business day slave side creation date issue.
None	16218	No	Sudhakar		Assurance	Submitted	Last Sale prefix of 'AUCT' should not be effective for cross product strategies.

8515	16511	no	Connie	Assurance	Submitted	Ops requested to to change email for future settlement reports.
8537	16856	Yes	Sudhakar	Assurance	Submitted	MKT order was not trading with incoming AR at a BETTER price than NBBO.
8523	16943	No	Khosravani	Assurance	Submitted	GUI Filter correction
N/A	N/A	No	Connie	Assurance	Submitted	Non Global Server Processes to use new JDBC driver as requested by Infra and agreed by Steve Sinclair
8551	17047	Yes	Hemant	Performance	Submitted	HANG-UP OBSERVED DURING BC FAILOVER WITH MULTIPLE TRADE SERVERS.
8563	17738	No	Montiel	Assurance	Submitted	Now using better penny price.
8564	17740	No	montiel	Assurance	Submitted	Send order to par if marketable with MQ.
	17750				Under Investigation	
	18402		Nikhil	Performance	Submitted	Corrected class consolidation
8667	19151	Yes	Manchana	Production	Submitted	No last sale when a carded trade is deleted
8368	19893		Gillund	Assurance	Submitted	recieving "getCHybridOrder" as "L"
8655	20450	Yes	Eck	Production	Submitted	

Ou2tage related enhancements

None

Outlier related enhancements

Performance related enhancements

Configuration and development/performance testing related enhancements

Other enhancements included in this release

The maxium order size set to 999,999 and the maximum equity leg set at 99,999,999.

Price Adjustments for OSI

Summary

Currently for leaps rollover and symbol change, in production orders are cancelled only for the adjusted products. With OSI, For Leaps all order should be cancelled for the class not just the adjusted products. Symbol Change, Orders will be cancelled for adjusted product set.

Test Setup

1. **setContext.main and setContext.process.OHServer** file has following changes
    - Introduced **-d** argument (***enableOrderCancellingAtClassLevel***) for Global server and respective OHServer (Hybrid, One_MAIN and CFE_MAIN)
    - By default enableOrderCancellingAtClassLevel is true (i.e it will work in new may of Leaps RollOver)
    - Change it to false and bounce GloalServer and Respective OHServer abd use ***pargs pid |grep –I enableOrderCancellingAtClassLevel*** to see that it has picked up new value (i.e false) in order to test the existing behaviour.
2. Please follow the existing Price Adjustments test plan.
3. Pick two product classes with more than one reporting class under each.
4. Use one product class to do Leaps and another product class to do Symbol change.
5. Enter GTC orders for Leaps and Symbol change.
6. For Orders entered by a Cancel firm,
    a. All orders for LEAPS should be cancelled.
    b. Orders for Symbol change should be cancelled only for the adjusted product set [Works as before]
7. For Orders entered by an Adjust firm,
    a. All orders for LEAPS and Symbol Change should not be cancelled.
    b. Orders are still in book that have entered by adjust firm next day.
8. Before applying adjustments,
    a. Verify the HD report reflects above-mentioned behavior for Order cancels.
9. After adjustments have been applied on CD at EOD. Please verify all orders have been cancelled for Cancel Firm and Orders are still in book that have entered by adjust firm next day.
10. In **production** we will need minor to trun off the flag to **false** when we rollout.

Start Session Changes

Summary

Currently when sessions are started we update session element which results in increase time to start session. We are using quick start option to start session for all session by default which bypasses updating the session elements.

Test Setup

3. **In setContext.main** file has following changes

- Introduced **-d** argument (***startSessionQuickFlag***) for Global server
- By default startSessionQuickFlag is true (i.e it will start session with quick flag ON)

4. When session are started by cron job every day Check by running following script to see that no classes are in NO_SESSION state , all of them must be in CLOSED state.

$RUN_DIR/***showProductStatesBySession***

Infra 13.1

Included in QAINFRA_13.1 (Infra 13.1) base release, which is baselined to Latest QAINFRA_13.

- Use JDK 1.6.0 Update 18
- Add millisecond precision to times in outlier monitor log
- Removed amqconn.setAlwaysSessionAsync(false). False value caused problems with GMD redelivery.
- Turned off ExpSecRef flag for backup ChannelAdmin. Change is in startChannelAdminWorker
- More JGrinder maintenance, SEDL 8433; 1. replaced HashMap with ConcurrentHashMap, and 2. made variable mi volatile.
- Add defensive bugfixes for parseACL_Services.pl
- fix an issue of ProcessWatcherProxyECimpl related to process states of initial query.
- (SEDL 8434) Change the defaults for RepeatWriteAttempts and WriteSleepTime from 100 and 100 to 1000 and 1 respectively.
- Holidays table maintenance; removed obsolete 2008/2009 entries & added 2013 entries
- ExtentMap change: BCDNs now pull in CacheUpdate from GCDN. Provides a fan-out effect to lessen the load on the GC AMQBroker.
- Fix DN_ARGS settings. Moved out the settings for disabling RT usage from prod settings so that they always get set
- fixed problem with createCalendarServiceEC: added infra interfaces to interface list
- Changed ActiveMQJmsAdministratorImpl so it would check if ChannelAdmin is available before it sends a query to get the active URL list of AMQBrokers. Also changed the type of log entries from sysAlarm to sysWarning if it cannot get active URL list, because they are really not essential
- AMQBroker performance improvement: updated AbstractRegion.java to reduce list contention / iteration durings the “addDestination” code path. Added some logging as well. Also fixed a problem with composite destinations that had wildcard component destinations.
- More Instrumentor Archiver perf enhancements:
  - Added earlier null check to avoid putIfAbsent() locks on a highly used method
  - Added date caching to File buffer.
  - Added date caching to NC and Q instrumentors for common dates.
  - Removed loop invariant SimpleDateFormat in NC and Q instrumentors, placing it in the cache
- Trader Service usability & performance improvements; includes SEDL 8432:
  - Improved logging
  - Impls now have begin and end messages around methods doing service API.
  - All exceptions are now caught and re-thrown after being logged; some are just logged as appropriate
  - Defined a logging builder to make standard logging output (easier to grep)
  - Removed redundant call to getServiceTypeByName() in withdraw_using_constraint; getTradeOffersForType() does the same check, so it is a needless DB action
  - Removed commented-out code in various classes; were some earlier Logging changes (from 2003) referring to APIs that no longer exist.
  - Added a cache for TraderServiceType objects; this will mainly improve the performance of TraderDBUtil.getServiceTypeByName() which is called by a substantial number of service methods.
  - Removed R/W lock from the TraderService top layer & pushed this lock down to TraderDBUtil. Locks are now much smaller, and not R/W in the case of TraderOffer. This will reduce writer contention during overall system start up.
- LogWatcher property updates for IDService (warnings), add CFIX directory structure, and a watch for CMS PreClean aborts.
- Fix startInfraSystem “IGC” mode to rearrange statup order
- Fix createCalenderServiceEC to create event channel with proper interfaces and a default to AMQ channel

- Create a new Outlier Instrumentor type and publish it on the Instrumentation Channel. It must also be persisted by the Archiver.
- Add support for OutlierInstrumentor in CommandConsoleUtility to allow display in cmdConsUtil.
- Updates to Infra ACL handling for better configuration management
- Fix an issue in CalendarService where the service was restricted to run only in the US/Central time zone. Refactor CalendarService so it will support any time zone
- Have the IGCDN publish Instrumentation to one common ICS Instrumentation extent name instead of one per ICS machine. Extents are set with FilterFlag=false, so individual extents are inefficient
- Channel Admin optimization that includes use of a serialized Extent Map objects to speed startup
- InfraEOD change to restore extentmap.ser file to log directory (for channel admin)
- AdminPropertyDefinition change to set the default channel admin and extent map admin server name to match the real configuration in most environments
- Update perf environment's genInput files
- Provide dynamic discovery of active ("master") AMQBrokers and transform the client's broker URL list accordingly.
- Added a feature to execute remote commands via cmdConsUtil. See the Infra Wiki for details of the changes.
- Added a remote method to the InstrumentationArchiver that will export only the IDs that have been used during runtime
- Update cmdConsUtil to ask "are you sure" for a Full GC directive
- cmdConsUtil maintenance: don't ask for confirmation when querying instrumentors; also corrected "usage" text
- Changed TopicConnectionFactoryActiveMQImpl::createTopicConnection so that it sets "alwaysSessionAsync" flag on the connection to *false*. There's no reason to have the extra SessionTask threads used for dispatching.
- Extent Map updates to add a "pull" of QRM channel for BCDN into the BC AMQ cloud to lessen the load of the global AMQBroker
- Support for TTE in pub/sub events
- Events TTE fixes:
  1) fix inconsistent reporting on the DN receiver side due to incorrect use of connection name where subject name is required
  2) add a check to prevent a JMSException being thrown silently when a TID is added to an event downstream on an event which already has a TID
  3) set SMS_ARGS in setContext script so SMSServer uses the global registrar; change is needed as SMSServer receives some app events & SMSServer has to be known by global registrar for proper ET reporting.
- Modify EOD scripts to "scrub" instrumentation IDs that change daily
- In Grizzly / IIOP, remove extra Selector objects per connection. Needed to replace code using that Selector object such that partial writes were handled and "timeout" conditions were maintained even if a socket was in a "stuck" (full) state for an elongated period of time.
- Optimized InstrumentorArchiver to significantly reduce CPU and memory consumption:
  - Replaced string concat keys with an object tuple
  - Replaced "+" operators with StringBuilder where appropriate
  - Opportunistic caching of stringified dates
- Update ActiveMQ base software to version 5.2 GA
- Add support for Master/Slave negotiator in AMQBroker. This will allow the Master broker to match up & run on the same "side" as other application processes
- In support of AMQBroker M/S Negotiator, have clients dynamically discover which is the "Active" AMQBroker (from a list of brokers) and arrange the Broker URLs accordingly
- Changed TopicConnectionFactoryActiveMQImpl::createTopicConnection so that it sets "alwaysSessionAsync" flag on the connection to *false*. There's no reason to have the extra SessionTask threads used for dispatching.

- Modifications were made to LifeLine processing to get the LifeLine reference from trader. This impacted RemoteProcessWatcherWatcher and LifeLine code. Also included changes to CAS & SACAS..setenv files to include a variable with PW names.
- A new tool called IORtool which can display and maintain the initrefs.ior file.
- Script createCalendarServiceEC was modified to include the CP_SERVER variable in the classpath
- Minor changes to the LogWatcher throttling mechanism to include the string associated with the throttle and to issue message when throttle triggered.

Implementation Plan 8.3 for C1

GC Database changes

Run the following in GLOBAL to drop the existing global index on table *recap_reports_archive* and *mkt_hist_reports_archive*, and create partition index instead:

```
whenever sqlerror continue
whenever sqlerror exit failure
alter table recap_reports_archive drop constraint recap_reports_archive_pk;
alter table mkt_hist_reports_archive drop constraint mkt_hist_reports_archive_pk;

create index recap_reports_archive_pk
on recap_reports_archive(dayofweek,databaseIdentifier) local;
create index mkt_hist_reports_archive_pk
on mkt_hist_reports_archive(dayofweek,databaseIdentifier) local;
```

BC Database changes

Installation procedures GC02A/B

QA Steps

At 3:15 have qa load the new software.

Server Group steps

Most of the steps here can be done after 3:15.

Only if needed fix the setContext file in /sbt/prod/tradeeng directory. After you login, if the setContext version has changed then the setContext is not run and you will not be able to start any process. In this case just copy the /sbt/prod/tradeeng/CBOEDIR_8.3/setContext.template file into /sbt/prod/tradeeng as setContext and then correct all the variables in it. You can use the old setContext file as an example to update the new file.

Change run_dir links for previous release.

Change the run_dir link in /sbt/prod/tradeeng to point to the new release.

Logout and log back in as tradengp.

Run script $RUN_DIR/bin/genWatchedProcessList. Verify that $RUN_DIR/properties/ WatchedProcessListServer.out is generated and that all processes are listed correctly in this file.

Do a diff against the old and new WatchedProcessListServer.out file to ensure they are the same. **DO NOT go any further if this does not work.**

Installation procedures GC01A/B

- **QA steps**

  1. At 3:15 have qa load the new software using the QA setup steps.

- **Server group steps after end of all sessions.**

  4. Install new software CBOEDIR_8.3 release,
  5. Update setContext in home directory with changes
  6. Change run_dir links and log back in.
  7. Do any database conversions if needed..
  8. Master side only. Create New Event Channel: createSbtEC TransientQuoteStatus
  9. Start the GC processes.
  10. Add new trading properties, event channel if any:
      - TradingPropertyServiceClient setPropertyDefinitions ${RUN_DIR}/properties/TradingPropertyDefinitions.csv
      - Execute ${RUN_DIR}/bin/initPMMTradingProperty –run to initialize defaults for AllocationStrategy for Complex Products to use the new configuration for PMM
      - Execute initQCTTradingProperties script to initialize QCC Minimum Volume for all classes.
        - This will give error when installing 8.3 for first time on a GC box, since the property will not be found on querying it. It can be ignored, it is a valid exception

      ```
      Displaying the current values.  Output is in file:
      /sbt/prod/tradeeng/run_dir/tmp/initQCTPropertiesDISPsettingsBefore_142622.txt
      Note. If the values were never set before (i.e. from a new TradingProperty), the
      getTradingProperty command will fail, which is expected.
      ----------------------------------------------------------
      processing session:  ONE_MAIN

      Found target service, getting command
      Invoke
      ProdGlobalGlobalServerA.GlobalTradingPropertyService.TradingPropertyServiceHo
      meLocalImpl(TradingPropertyServiceHomeLocalImpl).getTradingProperty(
      "QCCMinValue" "ONE_MAIN" "0" )
      Result:
      Results from "get":
      PropertyTradingPropertyService admin callback error:
      com.cboe.exceptions.NotFoundException:
      IDL:exceptions/NotFoundException:1.0 Trading property for
      key:QCCMinValue:ONE_MAIN:0 not found.
      ```

        o
  11. Enable global external connections on the GC.
  12. Start all sessions, do a quick quote and order test on one class on each bc.

13. Ops runs an “IPD Resync” report (Window staff would verify the data, a clean report is a good report)
14. End all the sessions.

▪ Other Verification after GC Upgrade

13. Check all files (.log, .debug, .out, .err) for errors, exception’s and high system alarms.

a. With PMM project, a new Allocation Strategy Element has been introduced, with 8.2 BC, HTS processes will have an exception on startup as below which is an expected behaviour.

```
high systemAlarm 2009/12/2 1:55:44.08 HybridTradeServer3
ProdBC20x1HybridTradeServer3sbtbc20a BC20x1
"AllocationStrategyFactoryImpl >>> Unknown or invalid priority
allocation strategy code skipped.  Code = 21" 0 (5 duplicates
suppressed, last created at 2009/12/2 1:55:44.08)
```

b.

14. If you are installing the Slave side box then perform a fail-over so that the upgraded box becomes Master and then continue on with the remainder of the plan.
15. Start all sessions using the SA GUI
16. Verify on prdgc01a/b that there are no products in NO_SESSION state.
17. Use SAGUI to open test products on all the BC’s. You can get the list of test classes from operations (This list is taped to one of the monitors in the basement)
18. Login to 2 trader GUI’s and bring up the market display window and status window on each of the GUI’s. NOTE: Status Window is a scrolling display that shows status messages for orders and quotes as and when they occur. (This window can be used to verify that OrderStatus and QuoteStatus events are working).
19. Do test trades using quotes/orders on at least one BC for all sessions.
20. For Futures TradesServers (TradeServer3) – You can use the same user to enter Orders/Quotes.
21. For Hybrid TradeServers (you will need to use quotes to trade with each other) + Enter a couple of orders.
22. Verify that CurrentMarket (Mkt Bid and Mkt ask) is showing up on the market display window. This tests out that current market events are working.
23. Verify that the last sale price and quantity fields are showing up on the Market display window. This tests out that Last Sale and Recap events are working.
24. Verify that Order Status messages (New Order, Order filled are showing up on the Status window screen)
25. Check all files on prdgc01 (.log, .debug, .out, .err) for errors, exception’s and high system alarms.

- **GC - Saturday verification after upgrade**

CBOEDirect verification after Global Cluster upgrade

#	Description	PASS/FAIL
	Perf truncate should not be run on BC97	
1	**GC01 - Verification test**	
2	loadOpenInterest test (performed by Ops at startup- requested at weekend test meeting)	
3	Product download test – CAS (performed by Ops at startup)	
4	Restart all CAS & Fix Engines. Verify CAS startup time. (performed by Ops at startup)	
5	Start all sessions using the SA-GUI (performed by Ops at startup)	
6	Verify all products are assigned to sessions and no product is in NO_STATE (Ops at startup)	
7	Run ITG Checkout scrpt on all BC's (performed by Ops at startup)	
8	Transition all products for all sessions to PRE-OPEN state.	
8.5	*** Special Test *** Verify in BC97 database there are resting SPX GTC orders. Failover BC97. **NOTE** do not suspend or halt classes before failing over. After failover put the OHS in debug: ar OHServerHybrid setLogLevel ALL DEBUG Page FIX support (Arun: 847-530-3493). Support will login a user (with GTC's) on FIX 50 engine 4, verify orders are received and canceled.	
9	Run the XTP replay, replaying the Friday 8:29-8:45 traffic at a 1.5 times rate. Verify that all W_MAIN classes transitioned to OPENING-ROTATION, verify ticker and recap on GUI, Verify underlying price in MDH. Verify broadcast to a PDS. Verifies Data from MDGC1.	
10	OPEN all products for all sessions.	
11	Kill a CAS to verify users are logged out - SMS test	
12	Enter Quote - Book Depth update (dynamic) - CFE/ONE_MAIN Verify that data goes ot of CfnAdapter1 on mdgc01 (3 outbound lines for CFE and 3 outbound lines for ONE). **ar cfnAdmin monitor raw CFE_MAIN C 1**	
13	Generate and verify Half Hourly reports for News Wire and HVOL. Actually sending the file to OPRA can only be verified on the weekend. Only on the weekend can you use the –opra option. * Verify hourly reports are generated every half an hour by Control-M * Run script **createHalfHourlyReport – nw W_MAIN**. * Verify that new report file is generated. log/HalfHourly_NewsWireReport_latest.xml.log * Repeat above for –hvol.	
14	Examine global log to verify open interest was loaded	
15	Using SA_GUI, spot check some open interest for some of the products. There should be non-zero values for most of the products. It is OK if a few of them have zero values because it might be a result of new series additions.	
16	Run **ar MarketDataReportServer1 showStatistics** and verify number messages received	
17	Do test trade on any production class for W_MAIN. Note : SATURDAY TESTING ONLY	
18	Repear step 16 and verify total number of events received increased	

19	Repeat step 13 and verify the reported numbers have increased	
20	Pause XTP Replay.	
21	Perform four BCXX failovers with GC01a is master. Where XX is a single hybrid, CFE, ONE, and STOCK BC.	
22	Run ITG Checkout scripts on the failed BCs (only). While running proceed to step 23.	
23	Verify that data CM , LS and Product states are going out to PDSs. The PDSs are verified as follows: Verify the PDS Overhead is updating with values from the class being used. Quote and last sale information should change. This verifies the MDB broadcast. Depending on the type of display OPS has configured on their test PDS Overhead, the last sale might not show up. Have them switch screen types until they find the one that shows the last sale. **Make sure XTP replay is NOT running during the ITG checkouts**. XTP Replay can cause spooling and there would be a possibility of dropped quotes/trades which could cause the test to appear to fail.	
24	Failover MDGC01 and MDGC02 with new GC01a as master.	
25	Verify CFE/ONE_MAIN book depth data is being published by repeating step 12.	
26	FE failover test with GC01a as master. KILL FE03	
27	Restart CAS2011. Verify the cas reinitializes. Run ITG Checkouts on any of the following BC/TE combinations that use FE03. BC04/TS-4 (cas2011) BC10/TS-3 (cas2011) BC30/TS-2 (cas2011) BC82/TS-1 (cas2011) BC90/TS-3 (cas2011) BC93/TS-4 (cas2011) BC98/TS-1 (cas2011)	
28	Start XTP Replay (1.5 rate(. Open the *Market Display For Underlying* window and verify ticker and recap. *Special Test* Note if broker on mdgc01 is discarding – if so then skip test 38	
29	VERIFY BROADCAST TO A PDS. THE PDSS ARE VERIFIED AS FOLLOWS:Verify the PDS Overhead is updating with values from the class being used. Quote and last sale information should change. This verifies the MDB broadcast. Depending on the type of display OPS has configured on their test PDS Overhead, the last sale might not show up. Have them switch screen types until they find the one that shows the last sale.	
30	GC01 Fail over (new to old)	
31	Pause XTP Replay.	
32	RUN CMi and FIX ITG Checkout scripts (all BCs) after the fail over.	
33	Verify Failover times to see how long it takes to do the complete failover (Stop + goMaster + Pre-open products)	
34	Close->PreOpen->Open all products in all sessions. Time the transition from Close to PreOpen	
35	Bounce a CAS to verify it “re-inits”	
36	Verify the CAS startup time against the previous morning startup time using the CAS Start Times script.	
37	Failover GC2.	

37.5	***Special Test*** Failback mdgc01. Reset to “old config”. Run xtp replay @1.5 rate and verify discarding happens.	**Skip**
38	Failback GC01 to CD 8.3 codebase.	
***	Set GC flag for OSI testing on. Bounce GC01 process. Begin OSI testing. Page/notify Client Support (FIX Support) to run OSI FIX tests on fix 50.01. Begin OSI Backoffice testing After OSI testing: Set GC flag for OSI testing OFF (Server Support). Set FIX OSI flag OFF (FIX Support).	
39	Close all products and run End Of Sale and updateClose after all other tests have completed.	
40	Run and verify any **new** end of day procedures as listed in the operator procedures	

Fallback

1. See operator procedures on how to failover GC’s.

General Settings to Turn on OSI Product Lookup.

- After 8.3 GC rollout testing, **Turn on the flag for OSI product Lookup** PROCESS_ARGS="${PROCESS_ARGS} -DenableOSIProductLookup=true"
- Once OSI testing is complete, **Turn off the flag for NON OSI Product Lookup**. PROCESS_ARGS="${PROCESS_ARGS} -DenableOSIProductLookup=false"

Installation procedure for the BC

- **QA steps**

1. At 3:15 have qa load the new software using the QA setup steps.

For the first, follow the *Evening Installation Procedures* below and then follow the *Saturday BC Verification After Upgrade* procedures on the following pages.

- **Evening Installation Procedures - Server group steps after end of all sessions.**

2. Verify that the BC table changes have been done as explained above. There are no BC table changes for this load.
3. Shutdown tradeengine on the Master BC (tradengp and tradengh [tradengh is only for W_MAIN] login).
4. Install new software CBOEDIR_8.3 release, change run_dir links .
5. Start the BC processes (tradengh [tradengh is only for W_MAIN]first and then tradengp logins).
6. Run the command **`businessExternalServices start`** to make sure that remote connections are established for the adapters on that bc.

- **Evening Installation Procedures – Verification during rollout**

7. Check all files (.log, .debug, .out, .err) for errors, exceptions and high system alarms.
8. If you are installing the *first* slave side box then perform a fail-over so that the upgraded box becomes Master and then continue on with the remainder of the plan.
9. Start the sessions affected by the particular BC install using the SA GUI (NOTE: some BC's can trade multiple sessions in different trade servers).
10. Verify on prdgc01a/b that there are no products in NO_SESSION state.
11. This instruction only applies if you are installing a W_MAIN BC. Run the following commands to get counts. These commands will be run again after ITG checkouts. The counts should increase.
    i. Run **ar MarketDataReportServer1 showStatistics** to get count
    ii. Run “hsAdmin –c stats –p HybridHistoryServer1” to get count
    iii. Run “`hsAdmin –c stats –p HybridTradeServer1`” to get count
12. Run ITG Checkouts
13. If installing on a BOB BC (If not installing on a BOB BC, skip this step), enter a manual quote and verify the current market is updated.
14. This instruction only applies if you are installing a W_MAIN BC. Repeat step 11.
15. Check all files on the affected BC's (.log, .debug, .out, .err) for errors, exceptions and high system alarms.

Note Some times after end of session the remote application will not allow us to connect to them. So the only way to verify is by looking at the log file to make sure that the program's are making an attempt to connect to the remote system on correct ip address and correct port nbrs.

- ## Final verification

16. Close all products in all sessions using the SA GUI (Pick the tab to close ALL the products).
17. Verify memory usage for **all processes** on the BC and Garbage Collection activity is stable using pstats. Or by comparing with the OLD and NEW .out files. **(STILL NEED TO DETERMINE WHAT PASS/FAIL IS)**
18. End all the sessions
19. If you started this upgrade on the Slave side perform a fail-over so that the upgraded box becomes Slave now.
20. Test Complete, Notify operations.

- ## Fallback

1. See operator procedures on how to failover BC's or list instructions specific to your release here.

- **Saturday BC verification after upgrade**

#	Description	PASS/FAIL
	Hybrid BC - Verification Test	
	You will need 2 trader gui's and 1 sa gui - login thru all these guis before you start the test. XtpReplay data should be captured on a production day between 8:29-8:45. This applies only if installing on a W_MAIN or W_STOCK BC.	
1	Pre-requisites – Ops has brought up system and successfully ran checkouts	
2	Verify on prdgc01a/b that there are no products in NO_SESSION state.	
3	Ask Ops to verify the “checkout results” e-mail that is sent to the CCS e-mail group when checkouts complete. There should be 3 e-mails for CMI and 3 e-mails for FIX.	
4	Verify XTP Data is being received by the trade servers, Underlying recap (W_MAIN), BOTR exchange indicators (W_MAIN and W_STOCK using a production class) - Verify using ar commands	
5	Using a Trader GUI, do MDH Queries and verify NBBO and exchange indicators.	
6	Run the following commands to get counts.These commands will be run again after ITG checkouts. The counts should increase. Run **ar MarketDataReportServer1 showStatistics** and note the number of messages received. Run “hsAdmin –c stats –p HybridHistoryServer1” to get count Run **“**`hsAdmin –c stats –p HybridTradeServer1`**”** to get count	
7	Run ITG checkouts for Trade Server 1 on the BC being installed. Make sure XTP replay is NOT running during the ITG checkouts. XTP Replay can cause spooling and there would be a possibility of dropped quotes/trades which could cause the test to appear to fail. **WHILE ITG CHECKOUTS ARE RUNNING** - Verify current market, last sale, trade reports, fill reports over all external connections. Enter a quote, order, and trade for a production class for the W_MAIN, W_STOCK, W_ONE, W_CFE sessions. OPS should verify: **CTMr** – verify there are no un-acked trade reports. See Operational procedures. **COPP** – This is for W_MAIN. verify the COPP GUI shows a queue being build for the OPRA lines. This will be true if COPP was brought up *without* OPRA being live and *without* the OPRA simulator being up. These conditions result in no place for COPP to send the data and it will queue inside COPP. **FOPP-ONE** – This is for W_ONE. verify the FOPP-ONE GUI shows an increase in the number of messages that were sent out the outbound lines. **FOPP-CFE** – This is for W_CFE. verify the FOPP-CFE GUI shows an increase in the number of messages that were sent out the outbound lines. **STOPP** – This is for W_STOCK. verify the STOPP GUI shows a queue being build for the outbound lines. This will be true if STOPP was brought up *without* STIC/NASDAQ being live and *without* the STOPP simulator being up. These conditions result in no place for STOPP to send the data and it will queue inside STOPP. **PDS** - This instruction only applies if you are installing a W_MAIN BC. Verify the PDS Overhead is updating with values from the class being used. Quote and last sale information should change. This verifies the MDB broadcast. Depending on the type of display OPS has configured on their test PDS Overhead, the last sale might not show up. Have them switch screen types until they find the one that shows the last sale.	

8 Repeat step 6 to verify counts increase.

9 If this is a BOB BC installation Operations or on site support staff need to enter a manual price report in for one of the products on the Trader GUI. The last sale column should be updated. Check, MDH, there should be an entry for this last sale.

10 **BC Failover test (BC 28 & BC 9)**

12 For the BC being installed, use the procedures as listed in the operator procedures and have ops execute the BC failover over procedures.

13 Time the fail over, it should take roughly 3 minutes

14 After the fail over run the ITG checkout script on all BC’s.

If installing on a BOB BC: (otherwise, skip this step)

Enter Manual Quote – Verify Current Market Update

Enter Manual Price Report – Verify Last Sale Update – Verify MDH entry

15 Open DSP (Display Price Series) screen and check information displayed for product. The DSP screen is opened on the Trade GUI using a login that has a *price reporting* roll.

18 Fallback to the new BC (if it is stayin in production service) and run ITG checkouts on all 4 trade servers for the new BC. Backout any BC that is not staying in for Monday.

19 Close the products, End the sessions and verfy that all the sessions have ended successfully.

Installation procedure for the FE

- **QA steps**

  1. At 3:15 have qa load the new software using the QA setup steps.

- **Evening Installation Procedures - Server group steps after end of all sessions.**

Note: FE is installed using script setContext.main(vobs\server\properties\). In 8.3 this script generates setenv(for GC, BC and FE) with jdk1.6 update 18. For 8.3 FE Rollout FE must use jdk1.6 update 14. Manullay rollback this change from setenv file to use jdk1.6 update 14.

```
export JAVA_HOME=/usr/local/jdk1.6.0_14
```

Installation procedures – SACAS hosts

QA steps

On installation day after end of trading, deliver planned INFRA and SBT releases and scripts to designated SACAS and SAGUI boxes.

Client group steps

Refer to ***CASProductionInstallation.doc*** and follow the implementation procedures.

(SACAS should already be configured to use JDK 1.6.0_14).

GUI group steps

Install SAGUI on one box (possibly 2 or 3 boxes) so that SACAS can be verified.

Verification

Verify that each SACAS rolled out is visible in Patrol.

Use new SAGUI to log on and ensure that it receives data from SACAS. In particular, check the User Management Window, the Product Class Groups window, and the Product Definition window.

On the first day of SACAS rollout, check all files (.log, .debug) for errors, exceptions and high system alarms.

Fallback

Follow the standard backout procedure.

Installation procedures – CAS hosts

QA steps

On installation day after end of trading, deliver planned INFRA and SBT releases and scripts to designated CAS boxes.

On installation day after end of trading, install new configureCAS.ksh onto each CAS box being upgraded.

Client group steps

Refer to ***CASProductionInstallation.doc*** and follow the implementation procedures.

This relase includes SEDL #SYS008258 changes for reading values of variable PW_NAMES from ~/.setenv. Please verify the correctness of PW_NAMES value.

This release includes SELD #SYS008255 changes. DN filter delay is configurable to any value from this release onward. If not configured the default value is 2000 ms. No special action is needed unless this value needs to be changed.

SEDL ticket SYS008259 is part of this release. No special actions needed.

Verification

Verify that each CAS rolled out is visible in Patrol.

If today's rollout included a floor cas (cas0014, cas0015, cas0016, cas3005), use a Trader GUI to connect to it and make a trade using a test symbol. Ensure that the MarketDisplay window uses MDX. If possible, do two tests, one with a product in session W_MAIN and one with a product not in W_MAIN.

On the first day of CAS rollout, Infrastructure group verifies that CAS is getting status via the ExternalQuoteStatus and ExternalOrderStatus channels.

On the first day of CAS rollout, check all files (.log, .debug) for errors, exceptions and high system alarms on the CAS host and on the appropriate FE pair.

Fallback

Stop CAS and Infra processes.

Client Group will change the run_dir and v2cas* directories to previous release.

Operations will start Infra and CAS processes via Patrol.

Installation procedures – FIXCAS hosts

QA steps

On installation day after end of trading, deliver planned INFRA and SBT releases and scripts to designated FIXCAS boxes.

On installation day after end of trading, install new configureCAS.ksh onto each FIXCAS box being upgraded.

Client group steps

Following additional settings are needed for FIXCAS rollout. Please append following lines to respective fix boxes under respective setContext files.

<u>setContext .v2fixcasNN changes</u>

Hybrid

```
export OSI_SESSIONS=W_MAIN
export FIX_USER_THREAD_SIZE=30
export FIX_OUTBOUND_USER_THREAD_SIZE=3
export CLIENT_BC_CONN_POOL_SIZE=20
export USER_ID_PATTERN="DD[0-9][0-9][0-9]|CD[0-9][0-9][0-9]"
```

CFE and ONE_MAIN

```
export FIX_USER_THREAD_SIZE=30
export FIX_OUTBOUND_USER_THREAD_SIZE=3
export CLIENT_BC_CONN_POOL_SIZE=20
export USER_ID_PATTERN="DD[0-9][0-9][0-9]|CD[0-9][0-9][0-9]"
```

STOCK

```
export FIX_USER_THREAD_SIZE=200
export FIX_OUTBOUND_USER_THREAD_SIZE=20
export CLIENT_BC_CONN_POOL_SIZE=100
export CLIENT_HEAP_SIZE=3100M
export USER_ID_PATTERN="TEST.*|B[0-9][0-9]+|M[0-9][0-9]+|FS[0-9]+|FS[A-Z]+|DD[0-9][0-9]+|CD[0-9][0-9]+"
```

<u>JDK Versions</u>

jdk_1_6_0_14 for Hybrid , CFE and ONE_MAIN
jdk_1_6_0_18 for Stock

<u>Fix back up boxes need to be changed for FIXSERVERPREFIX</u>

Whenever we install the FIX backup boxes we need to update this entry in setContext.v2fixcasNN file. If engine is fix50BK engine 1 then add 50 to the engine number and remove the BK and update the variable as shown below

Current
export FIXSERVERPREFIX=**PFIX50BK01A**
to
export FIXSERVERPREFIX=**PFIX5051A**

Verification

Verify that each FIXCAS rolled out is visible in Patrol.

On the first day of FIXCAS rollout, run a FIX script to connect to the FIXCAS and make a trade using a test symbol. If possible, do two tests, one with a product in session W_MAIN and one with a product not in W_MAIN.

On the first day of FIXCAS rollout, check all files (.log, .debug) for errors, exceptions and high system alarms.

Fallback

Stop FIXCAS and Infra processes.

Client Group will change the run_dir and v2fixcas* directories to previous release.

Operations will start Infra and FIXCAS processes via Patrol.

Installation procedures MDCAS hosts

QA steps

On installation day after end of trading, deliver planned INFRA and SBT releases and scripts to designated MDCAS boxes.

On installation day after end of trading, install new configureCAS.ksh onto each MDCAS box being upgraded.

Client group steps

???

Verification

Ensure that Infra System and MDCAS processes start successfully when Operations starts them.

On the first day of MDCAS rollout, check all files (.log, .debug) for errors, exceptions and high system alarms.

Fallback

Follow the standard backout procedure.

Installation procedures – CFIX hosts

QA steps

On installation day after end of trading, deliver planned INFRA and SBT releases and scripts to designated CFIX boxes.

On installation day after end of trading, install new configureCAS.ksh onto each CFIX box being upgraded.

Client group steps

???

Verification

On the first day of CFIX rollout, check all files (.log, .debug) for errors, exceptions and high system alarms.

Fallback

Follow the standard backout procedure.

Installation procedures LC's

QA steps

At 3:15 have qa load the new software using the QA setup steps as documented above.

Server group steps

Master or Slave side can be upgraded after 3:15. There is no need to wait for End Of Session.

If installing the master side box. Then shutdown tradeengine processes on the Slave Side first then shutdown tradeengine processes on the master side.

If installing the slave side then Shutdown tradeengine processes on the Slave side.

Verify QA setup steps from above (as listed in this document, use the check list for verification).

Only if needed fix the setContext file in /sbt/prod/tradeeng directory. After you login, if the setContext version has changed then the setContext is not run and you will not be able to start any process. In this case just copy the /sbt/prod/tradeeng/CBOEDIR_8.0 /setContext.template file into /sbt/prod/tradeeng as setContext and then correct all the variables in it. You can use the old setContext file as an example to update the new file.

Change run_dir links

NOTE : Delete orun_dir and move run_dir to orun_dir. Helpdesk needs "orun_dir" to look at old log files.

Change the run_dir link in tradeeng to point to the new release CBOEDIR_8.0

Logout and log back in as tradengp.

Do any database conversions if needed..

Have operations bring up tradeengine processes using PATROL

If this is the slave box then failover and run thru the verification steps listed below.

If you are installing the master side then run **`linkageExternalServices start`** to verify if the connections are established correctly.

Also use the OLA Fixometer to connect to OCC to verify if all connectivity is OK.

Note Some times because TPF is down and OCC is down you may be unable to connect to the external systems, in this case just verify in the log files that we have made an attempt and that the other systems are down at the time.

Verification

Check all files (.log, .debug, .out, .err) for errors, exceptions and high system alarms.

Make sure all initialization is complete on all processes.

If you are installing the slave side box then installation is complete. (Just make sure operations runs the slave box in master mode the next day).

When installing the Master side run through the procedure on "How to check Linkage processes using Test Orders". Follow the procedures as documented in the "Linkage Operator procedures". Basically a script "**`checkLinkageTestOrder`**" needs to be run that will test out the flow between all processes.

Verify memory usage for **all processes** on the LC's and Garbage Collection activity by comparing with the OLD and NEW .out files.