SEMOpx Information – 07 October 2020

M7 API – List of known issues, fixes and guidelines (schemas 6.x)

Dear API Customers,

As requested, we hereby provide you with this document listing the known open M7 API issues experienced over the past and proposing some guidelines clarifying the use of the M7 API Application.

This document should enable you to better implement your API Application, find information about fixes indicative timeline, put in place the suggested workarounds when relevant and remove them when the related fix has been released.

Please note that this list may not be exhaustive and concerns only the 6.0 schema version.

We kindly remind you that this document is provided to you for information and clarification purposes only. Therefore, it does not create any kind of new contractual obligation for SEMOpx.

We also remind you that according to the Trading Agreement (Article 10) and the General Conditions for the use of Market Data (Article 7) entered into with SEMOpx, you shall treat the information provided in this document as strictly confidential and refrain from disclosing it to any third party, except to authorized receiving parties, provided that they have committed to an equivalent confidentiality obligation.

You shall also refrain from using any Market Data contained in this document for any purpose out of the scope of the Operational Rules of SEMOpx (Article 4.3.1) and General Conditions for the use of Market Data (Article 8).

Should you experience any new API issue, we kindly invite you to check it first in this list. If the issue is not listed hereto or in case of any doubt, please contact us at info@SEMOpx.com.

This document will be updated from time to time and available on our website (in the member area) or using the link and password provided with the initial version (please contact info@SEMOpx.com if required):
https://www.marketdata.epexspot.com/index.php/s/KrcXu3Hx4HSF3Je/authenticate

Best regards,

SEMOpx Team

History table

Version	Date	Description
V2.0	22 May 2020	M7 6.9: • Fixed SERVICE-6241: UDB order: Contract ID received after the Order Book msg • M7 6.9 open issue EPEX_MT
V2.01	22 June 2020	M7 6.9 :SERVICE-2645 : not a bug
V2.1	16 September 2020	SERVICE-

Contents

1 List of M7 API issues

1.1 Fixes delivered with M7 6.10 in ASIM

1.1.1 SERVICE-3168 remote Order book revisionNo now starts at 1

Context	Order book messages feature an attribute called revisionNo with the following description in DFS 180 for M7 6.9: *This value is increased in case of any change in the order book.* *Please note: the revision numbers of the order books are stored in memory only (not persisted) on backend side. After a restart of the backend system, the revision numbers of the order books will start again from 0.* This attribute, common to Public Order Book responses and broadcast, is designed to functionally identify order books versions. It can potentially be discontinuous (and as a result should not be used to identify gaps between broadcasts) but should always increase (except if the system restarts).
Issue description	The first received Public Order Book Delta Report broadcast message related to an XBID contract opening for delivery tomorrow has in 6.9 a revisionNo = 2. Several customers wondered if this was the sign of a gap between 2 order book broadcasts. As reminded above, the revisionNo attribute is not designed to detect any gap between two broadcasts for the same routing key (RK), this is the role of the sequence number. Reminder: RevisionNo can be discontinuous and should not be used to detect gaps. For more information please check the DFS180 3.2.4 Sequence counting for Broadcast Messages and the dedicated API FAQ section in the API package. So if an app receives such a broadcast and its [sequence number] = [previous broadcast for the same RK sequence nb] +1, then you can be sure there is no gap. The same routing key means the same product/area combination: for instance all hourly contracts deriving from an hourly product share the same RK, meaning that indeed a previous order book broadcast has been received before by your app for another order book related to the same area and to another contract on which there was same order activity before, and which enabled to initialize a reference sequence number for that RK. But there was still a potential problem: in the situation where an API app does not have this “previous sequence nb” for a RK, for instance because it started between the last broadcast for this RK and the order book opening, it has no mean to determine if there was a gap before that

	broadcast when receiving it (responses to inquiry requests sent when an app starts do not include any sequence nb, but only revisionNo). As a result when an app receives a Public Order Books Delta Report related to the contract opening for this area with a revisionNo = 2, this "2" value is perceived as ambiguous: one may think that there was at least a previous broadcast that was missed.
Impacted messages	• PblcOrdrBooksResp, • PblcOrdrBooksDeltaRprt
Fix	RevisionNo or order books starts at 1 as of M7 6.10. New description in 6.10: *This value is increased in case of anfirst empty order book after contract gener y change in the order book. Starts from 1 (ation).* *Please note: the revision numbers of the order books are stored in memory only (not persisted) on backend side. After a restart of the backend system, the revision numbers of the order books will start again from beginning.* As a result, API apps are always sure that there was no gap (no missed order book broadcast between the app start and the 1st received broadcast).
Fixed in version	M7 6.10.202

1.1.2 SERVICE-5690: "Data access exception occurred" in response to Public Trades inquiry requests

Context	The *Public Trade Confirmation Request* enables API apps to retrieve public trades that were executed during a certain [Start date, End date] period.
Issue description	A few API apps would randomly experience a "Data access exception occurred" error when trying to retrieve public trades.
Impacted messages	• Public Trade Confirmation Request
Fix	New indices have been added to the M7 database to speed up public trades retrievals.
Fixed in version	M7 6.10.149

1.1.3 SERVICE-6186: Remote trades: unexpected state transitions and revisionNos in PblcTradeConfRprt after sending a TradeRecallReq request

Context	When sending a TradeRecallReq request on a trade done on XBID contracts, two outcomes can occur: - The counterparty of the recalled trade is EPEX: in that case, the recall request is automatically accepted. The final state of the recalled trade is "RGRA" (recall granted). - The counterparty of the trade is another exchange/NEMO: in that case, the recall is manually managed by the Market Operators. They have 20 minutes to accept or reject the request. After this period, the recall request is automatically rejected. In that case the final state of the order is "RREJ".

	As result the expected sequence of remote trades state is: ACTI->RREQ->RSFA->RGRA or ACTI->RREQ->RSFA->RREJ. On top of that trades RevisionNo is supposed to increase at each state change.
Issue description	Following a recall request PblcTradeConfRprt would: - Contain unexpected transitions such as RREQ->ACTI, - Miss the RSFA state - Be sent twice (ACTI->ACTI, or RREQ->RREQ) RevisionNo between broadcasts would be duplicates (2, 2) or even decrease (e.g. 2,1).
Impacted messages	• PblcTradeConfRprt
Fix	Expected behavior described in the “context” section above.
Fixed in version	M7 6.10.149

1.1.4 SERVICE-6420: No H2H heartbeat with sobConnectionState =DISCONNECTED when disconnected from XBID

Context	In DFS180, chapter 6.4.27, the sobConnectionState attribute of the HubToHubHeartbeat message should change from CONNECTED to DISCONNECTED when M7 disconnects from XBID: more precisely one “additional“ heartbeat message should be sent with this status: DFS180: "When M7 disconnects from XBID, the last heartbeat sent by the system contains sobConnectionState =DISCONNECTED.“
Issue description	M7 would not send the DISCONNECTED HubToHubHeartbeat message.
Impacted messages	• HubToHubHeartbeat
Fix	Heartbeats sendings have been fixed in the H2H module in case of a disconnection from XBID. After having detected a disconnection from the H2H module (30 second period) M7 sends a HubToHubHeartbeat with sobConnectionState = DISCONNECTED.
Fixed in version	M7 6.10.149

1.2 Closed issue with M7 6.9 in ASIM

1.2.1 EPEXMT-2645 Extra Order Execution Reports when automatically reconnecting to XBID: NOT A BUG

Context	There are two types of M7 disconnections/reconnections from/to the XBID system: manual (by market operations) and automatic (for technical reasons). In case of a manual or automatic reconnection M7 is expected to send the new order state only once per order, to reflect the transition from the new Unknown state as of 6.9 to the Hibernated state in most situations.
Issue description	At the moment M7 sends: - a sequence of Order Execution Reports: HIBE, UKNW, HIBE - instead of only HIBE

	This does not affect the final result: the order is Hibernated, but these extra messages are not necessary. Please note there is no issue when reconnecting manually.
Impacted messages	Order Execution Report for remote orders coming from the Unknown state.
Fix	Get only an Order Execution Report for the HIBE state when automatically reconnecting to XBID.
Fixed in version	NOT A BUG. The "unnecessary" Order Execution Reports messages were actually relevant and due to an automatic M7 disconnection from XBID. Indeed M7 had detected an XBID heartbeat gap right after its initial reconnection to XBID: the system was behaving as designed by putting orders in the UKNW state (disconnection from XBID) before putting them back into the HIBE state (reconnection to XBID).

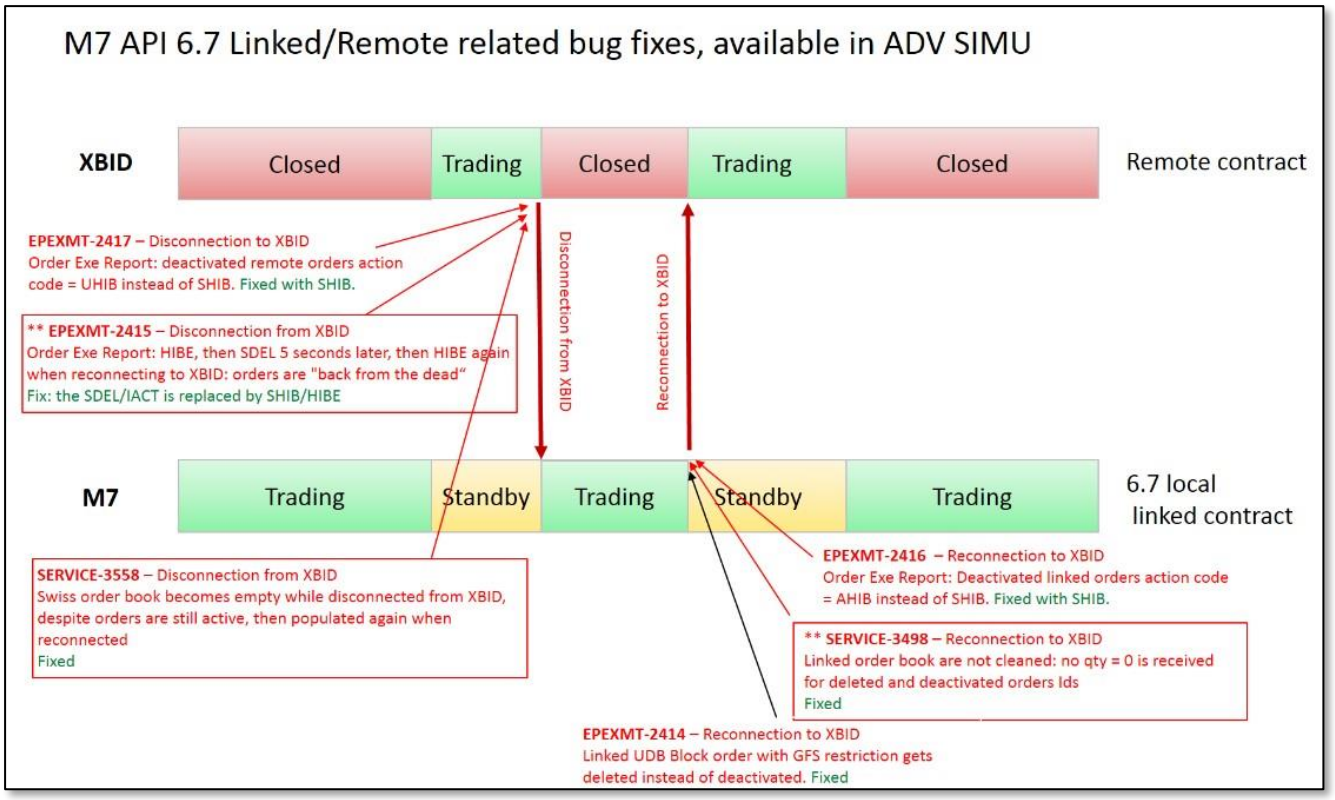
1.2.2 SERVICE-6241 Remote UDB Contract ID received after the Public order book message

Context	User Defined Block (UDB) contracts are created "on the fly" when the 1st order for this delivery period is being submitted. As a result: - a *Contract Info Report* is broadcasted by M7 to publish the new Contract ID - a *Public Order Books Delta Report* for this contract ID/Area combination is broadcasted, containing the UDB order
Issue description	The *Public Order Books Delta Report* arrived before the *Contract Info Report*: as a result, the contract ID was unknown and forced client API apps to send a Contract Info request. This issue was detected only for remote/XBID UDB Contracts.
Impacted messages	• *Contract Info Report* • *Public Order Books Delta Report*
Fix	For this specific contract ID creation context, the PublicOrder Book Delta Reports sending is kept by M7 until the Contract Info Report is broadcasted.
Fixed in version	M7 6.7.105 already in PROD This issue is related to the trading limit update done for all members at 16:00 CE(S)T.

1.3 Fixes delivered with M7 6.7 in PROD

1.3.1 Issues related to the switch between Local Linked contracts and Remote contracts (when disconnecting and reconnecting to XBID)

All the M7 6.7 issues in the diagram below are fixed as of M7 6.7.108.
Please note that this diagram becomes obsolete with M7 6.9 and the introduction the new Unknown Order State. The new diagram is available in our 27 May 2020 DatanewZ.



Precisions about the EPEXMT-2415 fix:

- Though required to avoid receiving a false order deletion, **the fix will affect applications which systematically try to delete or manage remote hibernated orders during XBID disconnections**. If this is your case, please adapt your logic, for instance by adding a criteria to check the tradability of the related remote contract before triggering that deletion behavior (or any other order management action). The deletion can only be done once M7 is reconnected to XBID and XBID contracts are tradable again.

- Indeed, when M7 disconnects from XBID (also called SOB for Shared Order Book), remote orders get hibernated.
- While M7 is disconnected from XBID it is functionally not possible to manage a remote (hibernated) order since the XBID contract is hibernated (state="HIBE" tradingPhase="CLSD").
- If you try to delete, hibernate, activate or modify such a HIBE order the operation will be rejected by M7 and the API will broadcast the following error message:
  a) If trying to delete the order: err="Order xx should be ACTIVE or HIBERNATED. The current order state is SOB_NOT_AVAILABLE." errCode="1089"
  b) If trying to modify the order: "Activation of remote order has been rejected because its state is unknown" errCode="1159"
  c) If trying to activate the order: "Product XBID_xx has been deactivated" errCode="2023"
- Messages a) and b) require a clarification:
  - When M7 disconnects from XBID, all remote orders are updated with an "unknown" status, called "SOB_NOT_AVAILABLE".
  - Before this fix, this SOB_NOT_AVAILABLE status was mapped to the IACT state (an Order Execution Report SDEL/IACT was sent).
  - Now this internal status is mapped to SHIB/HIBE so that apps do not believe the order is deleted.
  - These error messages just reflect this internal/external difference. We will ask for clearer messages in a further post go live release.

1.3.2 SERVICE-3550: a few API users do not receive any broadcast message

Context	All API users receive broadcast messages, including M7 heartbeats.

Issue description	All of a sudden, a few customers report that their API user can successfully log in but does not receive any broadcast message at all, including M7 heartbeats.
Impacted messages	All M7 API broadcast messages
Fixed in version	M7 6.7 already in PROD

1.3.3 SERVICE-3159: heartbeat loss followed by reconnection difficulties at 16:00

Context	We recommend API apps to consider M7 heartbeat is lost after 3 missed heartbeats.
Issue description	A few users experience a heartbeat loss just after 16:00, and then have difficulties to reconnect (can take several minutes).
Impacted messages	• M7 Heartbeat loss (more than 3 in a row) • Connection to the AMQP server takes a long time (30s) • Login Request do not get any response
Fixed in version	M7 6.7.105 already in PROD This issue is related to the trading limit update done for all members at 16:00 CE(S)T.

1.3.4 SERVICE-3203: heartbeat loss followed by reconnection difficulties at other times than16:00

Context	We recommend API apps to consider M7 heartbeat is lost after 3 missed heartbeats.
Issue description	A few users experience a heartbeat loss at different times than 16:00, and then have difficulties to reconnect (can take several minutes).
Impacted messages	• M7 Heartbeat loss (more than 3 in a row) • Connection to the AMQP server takes a long time (30s) • Login Request do not get any response
Fixed in version	M7 6.7.105 already in PROD The fix consists in dedicating to M7 heartbeats a 7th independent connection between M7 and the AMQP broker (nothing to be done on client application side). Please refer to DFS180 section 3.2.5 “Broadcast Distribution from M7 Back-End to AMQP Server” for more details.

1.4 Fixed issues with XBID 2.0: XBID Trading

1.4.1 Issue SMXBID-815: OrdrExeRprt for remote contracts: UDEL and UADD come in separate messages

Context	When an order management action gets processed by the M7 back end an *Order Execution Report* message is broadcasted by M7. When performing an order action which leads to a change of priority, the M7 API deletes the order and recreates it, leading to the actions UDEL and UADD, expected in the same *Order Execution Report* message.
Issue description	There are situations where UDEL and UADD are broadcasted in different Order Execution Reports. **Example when modifying the price of an order (112811529) on a remote contract without any matching:** • Message #1 for UDEL: <?xml version="1.0" encoding="UTF-8"?> <**OrdrExeRprt** xmlns="http://www.deutsche-boerse.com/m7/v6"> <StandardHeader marketId="EPEX"/> <OrdrList> <Ordr ordrId="112811529" initialOrdrId="112811529" clearingAcctType="P" acctId="TESTBG1---------" contractId="10553913" side="SELL" px="3773" qty="800" initialQty="800" ordrExeRestriction="NON" txt="" dlvryAreaId="10YDE-VE-------2" clOrdrId="a76c1cd7-45d1-4cd2-8234-2ffbd2c1fdab" preArranged="false" type="O" state="IACT" usrCode="TRD001" revisionNo="2" timestmp="2018-08-16T15:08:54.292Z" validityDate="2018-08-16T19:30:00.000Z" validityRes="GFS" action="UDEL" lastUpdateUsrInfo="TMEPEX-BG1-----XXBEPEX" counterOrder="false" remoteOrdrId="1992425" lastUpdateTm="2018-08-16T15:09:19.658Z"/> </OrdrList> </OrdrExeRprt> • Message #2 for UADD: <?xml version="1.0" encoding="UTF-8"?> <**OrdrExeRprt** xmlns="http://www.deutsche-boerse.com/m7/v6"> <StandardHeader marketId="EPEX"/> <OrdrList> <Ordr ordrId="112811530" initialOrdrId="112811529" parentOrdrId="112811529" clearingAcctType="P" acctId="TESTBG1---------" contractId="10553913" side="SELL" px="3775" qty="800" initialQty="800" ordrExeRestriction="NON" txt="" dlvryAreaId="10YDE-VE-------2" clOrdrId="a76c1cd7-45d1-4cd2-8234-2ffbd2c1fdab" preArranged="false" type="O" state="ACTI" usrCode="TRD001" revisionNo="1" timestmp="2018-08-16T15:09:19.620Z" validityDate="2018-08-16T19:30:00.000Z" validityRes="GFS" action="UADD" lastUpdateUsrInfo="TMEPEX-BG1-----XXBEPEX" counterOrder="false" remoteOrdrId="1992426" lastUpdateTm="2018-08-16T15:09:19.662Z"/> </OrdrList> </OrdrExeRprt> This is an XBID issue and the exact triggering conditions are still under analysis.
Impacted messages	OrdrExeRprt
Fixed in version	XBID R2.0 (in production since 31 October 2019)

1.4.2 Issue SMXBID-816: OrdrExeRprt for remote contracts: UDEL and UADD come in a single message but in the wrong order

Context	When an order management action gets processed by the M7 back end an *Order Execution Report* message is broadcasted by M7.

	When performing an order action which leads to a change of priority, the M7 API deletes the order and recreates it, leading to the actions UDEL and UADD, expected in the same *Order Execution Report* message. The UADD leads to a higher Order ID than the one deleted by the UDEL action.
Issue description	There are situations where UDEL and UADD are broadcasted in a single Order Execution Report but in the wrong order: UADD then UDEL. **Example when modifying the price of an order (112811529) on a remote contract so that the order gets fully traded against an existing buy order**: <?xml version="1.0" encoding="UTF-8"?> <OrdrExeRprt xmlns="http://www.deutsche-boerse.com/m7/v6"> <StandardHeader marketId="EPEX"/> <OrdrList> <Ordr ordrId="112811584" initialOrdrId="112811529" parentOrdrId="112811530" clearingAcctType="P" acctId="TESTBG1---------" contractId="10553913" side="SELL" px="0" qty="800" initialQty="800" ordrExeRestriction="NON" txt="" dlvryAreaId="10YDE-VE-------2" clOrdrId="a76c1cd7-45d1-4cd2-8234-2ffbd2c1fdab" preArranged="false" type="O" state="ACTI" usrCode="TRD001" revisionNo="1" timestmp="2018-08-16T15:13:45.406Z" validityDate="2018-08-16T19:30:00.000Z" validityRes="GFS" action="UADD" lastUpdateUsrInfo="TMEPEX-BG1-----XXBEPEX" counterOrder="false" remoteOrdrId="1992479" lastUpdateTm="2018-08-16T15:13:45.465Z"/> <Ordr ordrId="112811530" initialOrdrId="112811529" parentOrdrId="112811529" clearingAcctType="P" acctId="TESTBG1---------" contractId="10553913" side="SELL" px="3775" qty="800" initialQty="800" ordrExeRestriction="NON" txt="" dlvryAreaId="10YDE-VE-------2" clOrdrId="a76c1cd7-45d1-4cd2-8234-2ffbd2c1fdab" preArranged="false" type="O" state="IACT" usrCode="TRD001" revisionNo="2" timestmp="2018-08-16T15:09:19.620Z" validityDate="2018-08-16T19:30:00.000Z" validityRes="GFS" action="UDEL" lastUpdateUsrInfo="TMEPEX-BG1-----XXBEPEX" counterOrder="false" remoteOrdrId="1992426" lastUpdateTm="2018-08-16T15:13:45.465Z"/> <Ordr ordrId="112811584" initialOrdrId="112811529" parentOrdrId="112811530" clearingAcctType="P" acctId="TESTBG1---------" contractId="10553913" side="SELL" px="0" qty="0" initialQty="800" ordrExeRestriction="NON" txt="" dlvryAreaId="10YDE-VE-------2" clOrdrId="a76c1cd7-45d1-4cd2-8234-2ffbd2c1fdab" preArranged="false" type="O" state="IACT" usrCode="TRD001" revisionNo="2" timestmp="2018-08-16T15:13:45.406Z" validityDate="2018-08-16T19:30:00.000Z" validityRes="GFS" action="FEXE" lastUpdateUsrInfo="TMEPEX-BG1-----XXBEPEX" counterOrder="false" remoteOrdrId="1992479" lastUpdateTm="2018-08-16T15:13:45.465Z"/> </OrdrList> </OrdrExeRprt> A potential workaround is to sort the Order records within one OrdrExeRprt by (ordrId, revisionNo), so that the oldest orders (with the UDEL) are always processed first by your API application. This is a XBID issue and the exact triggering conditions are still under analysis. Therefore, it is still possible that there are situations where the proposed workaround enhances the sequence without making it 100% correct.
Impacted messages	OrdrExeRprt
Fixed in version	XBID R2. 0 (in production since 31 October 2019)

1.5 Pending issues: General

1.5.1 SERVICE-5227: unexpected logout reports for “inactivity” reason

Context	M7 can send Logout reports to a logged in user in various circumstances, because kicked out by someone else using the same user and logging in (with force = true), or for technical reasons.

	API apps must react to the Logout report by closing the AMQP connection, reconnect and try to log in again if relevant.
Issue description	A couple of customers receive spontaneous Logout reports from M7 because of "INACTIVITY" (text in the Logout Report message). This may be followed by difficulties to log in again for several minutes. So far, the only valid reason for which the login session is closed for inactivity is whenever a channel is closed while connection and other channels still exist.
Impacted messages	Logout Report
Fixed in version	Under investigations.

2 Guidelines: List of M7 API good-to-know behaviors

2.1 General

2.1.1 Change EPEXMT-2186: Removal of optional field exGTD in OrdrEntry

Context	In the Order Entry section of the DFS180, the field exGTD is set to "NOT to be used by trader".
Change description	Since this field is not required for traders and can be confusing it will be removed.
Impacted messages	OrdrEntry
Fix Version	The version in which the attribute will be deprecated is not yet determined.

2.1.2 Behavior SERVICE-5249: 2 AckResp in response to a ChgPwdReq

Context	The ChgPwdReq message enables API applications to change the connected API user password.
Behavior description	Sending this request, M7 sends: • A 1st AckResp in the private response queue if the Change Password request was correct XML wise, • An AMQP shutdown signal with the PWD_CHANGE reason when the password change is successful, • And a 2nd AckResp as a broadcast, sent before or after the shutdown signal, when the change is successful. The 2nd Ack was not described in DFS180: the next version will contain this precision. (done in the M7 6.10 documentation).

	When receiving the PWD_CHANGE shutdown signal API applications must close the current AMQP connection immediately and recreate a new one with the new password. The API user is logged out and needs to log back in after having recreated an AMQP connection.
Impacted messages	ChgPwdReq

2.2 XBID Trading

2.2.1 Behavior EPEXMT-1999: Remote contract orderbooks: totalQty calculation

Context	For each contract / area, each time a trade is executed, the total quantity featured in the orderbook is recalculated.
Behavior description	For a trade on a remote contract (XBID) between 2 orders: **1)** of the same market area (same or different delivery area): totalQty = [Trade qty], instead of 2* [Trade qty] **2)** different market areas: totalQty = [Trade qty], as expected Note: XBID project is studying the possibility to get this behavior aligned with local contracts behavior.
Impacted messages	Public Order Books Response, Public Order Books Delta Report

2.2.2 Behavior SMXBID-794: Disabling MsgRprt for IDCB Auctions

Context	When a capacity increase leads to a crossed order book (best Bid equal or higher than best Ask), an IDCB auction algorithm is triggered, executing trades up to the available capacity in order to uncross the order book.
Behavior description	Before XBID, when M7 was connected to the ICS platform, public messages were broadcasted at the start and finish of each IDCB auction, only if an auction had actually been executed. Since the introduction of XIBD the rule is slightly different: messages are broadcasted even if eventually no auction was executed (only the possibility of an auction was examined by XBID), resulting in a significant number of additional messages. These messages will be disabled in SIMU and then in PROD environments: done (on 30.10.2019 for PROD).
Impacted messages	MsgRprt

2.2.3 Behavior EPEXMT-2031: Contract Info Report only shows the current phase

Context	A contract can go through different phases in its life cycle, for instance: • In Germany for local contracts: Closed -> (local) Continuous Trading -> Closed (during XBID phase) -> (local) Continuous Trading -> SDAT -> Closed
Behavior description	The Contract Information Report only shows the current phase for each delivery area: it does not allow to anticipate what will be the start or end trading time of the phases to come. This enhancement is being studied by our provider.
Impacted messages	ContractInfoRprt

3 Releases indicative delivery timeline

	ASIM	**PROD**
Current Version on 16 September	M7 6.9.100	M7 6.9.100 + XBID 3.0
New version as of 18 September	M7 6.10.217	

M7 Version	**Delivery in ADV SIMU**	**Delivery in PROD**
M7 6.6	31January 2019	12 March 2019
M7 6.7.108	19 August 2019	10 September 2019
XBID 2.0	ELTS SIMU mid October 2019	31 October 2019
M7 6.8	5 Nov 2019 – technical delivery for configuration/checks 12 Nov 2019 – member test phase start	11 February 2020
M7 6.9	Q2 2020 – technical release + Unknown status	15 July 2020
M7 6.10	18 September 2020 (M7 6.10.217)	10 November 2020 (TBC)

Note: all provided delivery dates are subject to successful internal testing.

Best regards,

SEMOpx