



K2/ifsc

MMTS II data types

Prepared in 2001 by Effice Kft., based on the
authorisation of the Budapest Stock Exchange

For internal use only, by the Budapest Stock Exchange and its members

Version 1.5

The K2/ifsc program was prepared by Effice Kft.
E-mail: <effice@effice.hu>

Contents

1.	<i>Data types of the MMTS II trading system</i>	<i>3</i>
2.	<i>Values of the MMTS II new order fields</i>	<i>8</i>
3.	<i>Values of the MMTS II order amendment fields</i>	<i>9</i>
4.	<i>Values of the MMTS II order withdraw fields</i>	<i>9</i>
5.	<i>Values of the MMTS II Tick up/Tick Down fields</i>	<i>11</i>
6.	<i>Values of the MMTS II Market Maker order fields.....</i>	<i>12</i>
7.	<i>Rules related to MMTS II Order Types.....</i>	<i>13</i>

Index of modifications

Version	Author	Date	Summary of modifications
1.1	Effice Kft.	2001	First version
1.2	BSE	04/02/2005	Revised version Market Maker and Tick up/Tick down function fields added
1.3	BSE	04/02/2005	Corrected version Meaning of bit 28 in Ordermethods corrected
1.4	BSE	06/12/2017	Introducing new fields in connection of MIFID-2 regulations
1.5	BSE	07/06/2017	Expanding the Tick Up/Tick Down transaction with the new field

1. Data types of the MMTS II trading system

FixReal E.g. Price, yield, value. A two-element structure:

value: Value¹. Double. (Double precision floating point number)

decimals: Number of decimals, represented by a decimal integer value. This defines to how many decimals the number should be rounded off during display and calculations.

If during an order entry we do not wish to give any values in the FixReal field, then we should put NOT_DEFINED into the decimals field. (0.0 counts as a defined value.) When entering an order, the value of the decimals field is not interpreted by MMTS. It only represents information for the client querying the order entry board (e.g. to how many decimals the number should be rounded off).

SecBoardId A security identifier (MMTS terminology refers to it as a “*securities board*”). The trading method of a given *Security* on a given *board*. Its form: <BoardId><SecId>². The length of the *BoardId* is always BOARDID_LEN, while that of the *SecId* is SECID_2_LEN, and there are no separating zeroes between the two. The length of the data of *SecBoardId* is BOARDID_LEN + SECID_2_LEN which in the case of the MMTS II system is identical to the IFS_SECBOARDID_LEN. In the case of order requests, the *BoardId* and the *SecId* must be defined separately. However, when the securities board (SecBoard) is queried, the compound *SecBoardId* is given.

Date A date with a YYYYMMDD format, represented by a decimal integer value.

Time Time with a HHMMSS format, represented by a decimal integer value.

OrderMethods 32 characters, each either a '0' or a '1'; '0' is given when the given characteristic cannot be or has not been set, and '1' is given when it can be or has been set.

0. Market
1. Limit
2. Scaled
3. Negotiated Deal (Fixed-order)
4. Cross-agency deal
5. Market maker
6. Implied
8. Not published
9. BuyIn
10. Yield
11. Quantity
12. Stop
13. Settlement day

¹ As opposed to the MMTS I system, without transformation.

² As opposed to the MMTS I system, here the first character is not zero.

- 14. No trading account
- 15. FO Details
- 16. Fill All or None
- 17. Fill Minimum
- 18. Fill Any portion
- 19. Best Price
- 20. Order withdrawal upon logoff
- 23. Good Till Time
- 24. Session order
- 25. Day order
- 26. Good Till Date
- 27. Good Till Cancel
- 28. Fill and Kill (after filling any portion)

BuySell Buy/Sell. An enumerated type.

BuySellVerb_Buy Buying.

BuySellVerb_Sell Selling.

BuySellVerb_BuyOrSell Buying or selling meaning both sides (only in the case of a withdrawal, Tick up/Tick down or Market Maker Order Entry) .

OrderType Order types. An enumerated type.

OrderType_Limit Limit.

OrderType_Market Market.

Duration Expiry types. An enumerated type.

Duration_Immediate Immediate.

Duration_Session Session.

Duration_Day Day.

Duration_GoodTillDate Good till date.

Duration_GoodTillTime Good till time.

Duration_GoodTillCancelled Good till cancelled.

MMTS_Bool Logical values. An enumerated type.

MMTS_False False.

MMTS_True True.

Positiontype Position types. An enumerated type.

PositionType_OpeningTrade Opening.

PositionType_ClosingTrade Closing.

PositionType_DayTrade Daytrade.

BoolOp Relation between the values of the fields given in the withdrawal. An enumerated type.

BoolOp_And Logical *and* relation.

BoolOp_Or Logical *or* relation.

BoolOp_Not Logical *negation*. At present, it is now allowed.

CompareOp Types of comparison upon withdrawal. An enumerated type.

CompareOp_EQ Equal.

CompareOp_NT Not equal.

CompareOp_GT Greater than.

CompareOp_GE Greater or equal.

CompareOp_LT Less than.

CompareOp_LE Less than or equal.

Status Status. An enumerated type.

Status_Active Active.

Status_Suspended Suspended.

SecClassId Security classes. An enumerated type.

SecClassId_Index Index securities.

SecClassId_Spread Spread securities.

SecClassId_MultiLeg Multileg.

SecClassId_Debt Credit-type securities.

SecClassId_Equity Shares.

SecClassId_Future Futures securities.

SecClassId_Option Options securities.

SecClassId_Commodity Commodities.

NegDealInit Fixed transaction initiator. An enumerated type.

NegDealInit_Buy Buy-side.

NegDealInit_Sell Sell-side.

NegDealInit_Either Either side.

OptionVerb Put/Call. An enumerated type.

OptionVerb_Put Put option.

OptionVerb_Call Call option.

SettlementPriceType Settlement price types. An enumerated type.

SettlementPriceType_Indicative Indicative.

SettlementPriceType_IntraDay Calculated immediately.

SettlementPriceType_EndOfDay End of day.

SettlementPriceType_Final Final.

FirmClass Firm classes. An enumerated type.

FirmClass_Broker Broker.

FirmClass_Issuer Issuer.

FirmClass_Institution Institution.

FirmClass_Exchange Stock exchange.

FirmClass_Other Other.

CloseOutStatus “Close out” status. An enumerated type.

CloseOutStatus_CloseOut Close out.

CloseOutStatus_None None.

OrderStatus Order statuses. An enumerated type.

OrderStatus_OPEN Open.

OrderStatus_AMENDED Modified.

OrderStatus_MATCHED Matched.

OrderStatus_WITHDRAW Withdrawn.

OrderStatus_UNCOMFIRMED Unconfirmed.

OrderStatus_UNAPPROVED Unapproved.

OrderStatus_EXPIRED Expired.

OrderStatus_INACTIVESTOP Inactive stop.

OrderStatus_UNPLACEDSTOP Unplaced stop order. (not put into OB)

OrderStatus_UNPLACED Unplaced. (Not put into Order Book)

OrderStatus_EMBARGOED Embargoed.

OrderStatus_PRIVATEORDER Private order.

FillType Handling of minimum quantity. An enumerated type.

FillType_FillAny A part order (fill any portion).

FillType_FillMinQty Minimum quantity given.

FillType_FillOrNone All or None.

TradeStatus Trade statuses. An enumerated type.

TradeStatus_Matched Matched.

TradeStatus_WithDrawn Withdrawn.

TradeStatus_Unapproved Unapproved.

TradeStatus_Unconfirmed_buy Unconfirmed buy.

TradeStatus_Unconfirmed_sell Unconfirmed sell.

DissemStatus Disclosure statuses. An enumerated type.

DissemStatus_NotMatched Not matched.

DissemStatus_NoDelay No delay.

DissemStatus_Delayed Delayed.

DissemStatus_Sent Sent.

DissemStatus_Released Released.

DissemStatus_Lapsed Lapsed.

DispMode Display mode. An enumerated type.

DisplayMode_NODISPLAY Not to be displayed.

DisplayMode_STATUSLINE To be displayed in the status line.

DisplayMode_POPUP. To be displayed in a popup window.

EntityToWhatToWhom Type of the entity addressed or referred to. An enumerated type.

Entity_NONE None on the list below.

Entity_BOARD Board.

Entity_SECURITY Security.

Entity_SECBOARD Security board pair (security traded on that board)

Entity_TRADE Trade.

Entity_ORDER Order.

Entity_USER User.

Entity_FIRM Firm (brokerage house).

Entity_INSTRUMENT Instrumentum.

PriceParamArray It is a string type field.

This field defines the price step (tick size) for a given security & board pair. The tick size can be defined in ranges, and additionally it can be defined as an exact value or as a percentage of the order's real price. The price parameter structure is as follows:

$L|V|P_{dec}|n:m,[n:m], \dots$

Where:

- L is "P" or "Y" and indicates whether the lookup range refers to a price or a yield;
- V is "D", "P" or "Q", indicating whether the parameter value defaults to price/yield (as per the first parameter), or is a percentage or quantity;
- Pdec is a single digit integer defining the number of decimal places (pricedecimals) for price values in general.
- [n:m] are pairs of values that make up the array where n is the starting price/yield of the lookup range and m is the parameter value that applies to the range

The first range always starts at zero and ends at the start of the following range. The last range ends at infinity.

e.g. P|D|2|0:0.015,1.50:0.05

Meaning a price based rule where the parameters for the price ranges are:

Range ≥ 0 and $< 1.50 = 0.015$

Range $\geq 1.50 = 0.05$

2. Values of the MMTS II new order fields

The order entry record fields may receive the following values when a new order is entered³:

[TrdAccId] The character ID of the trading account.

[ExecutionId] The Id of the executor (user or algorithm)

BuySell BuySell type.

OrderType Order type.

Duration Expiry type.

PurgeOnLogoff Purge upon logoff.

AllowSoftQtyLimit Must be set to MMTS_TRUE.

AllowSoftPriceLimit Must be set to MMTS_TRUE.

[PositionType] Position type.

IsPrivate Private order. At present, it can only be an MMTS_FALSE.

[BoardId] The character ID of a securities board. Its value can be found in the *BoardId* field of the security record.

SecId The character ID of a security. Its value can be cut from the *Id* field of the security record.

[Price] Price, FixReal.

[Yield] Yield, FixReal.

Quantity Quantity, a positive integer.

[VisibleQuantity] Visible quantity, a positive integer.

[BrokerRef] A character field which can be completed by the broker freely.

[ExpDate] MMTS Date type.

[ExpTime] MMTS Time type.

[TriggerPrice] Activation limit, FixReal.

[TradeRef] Trade reference, for trade allocation.

[MinFillQty] Minimum quantity, a positive integer.

[InternalRef] A character field which can be freely completed, it will not be sent to the MMTS.

Either the `Price` or the `Yield` field may be defined, but they are mutually exclusive. The value of the `FixReal decimals` field will not be interpreted by MMTS. It only represents information for the client querying the order entry board (e.g. to how many decimals the number should be rounded off).

³ The square brackets imply non-mandatory data. If these are not given, then in the case of an INT type, we put in the NOT_DEFINED value, while in the case of a STRING type, we put in spaces.

3. Values of the MMTS II order amendment fields

The order entry record fields may receive the following values when an order is modified:

OrdNo The order number of the order to be modified. Its value can be found in the *OrdNo* field of the order record.

[OrdNoSpeedIdx] An internal index of the order in the MMTS trading system. (With the help of this, MMTS can find the requested order record more quickly.) Its value can be found in the *OrdNoSpeedIdx* field of the order record.

[TrdAccId] The character ID of the trading account.

[ExecutionId] The Id of the executor (user or algorithm)

Duration Expiration type.

AllowSoftQtyLimit Must be set to MMTS_TRUE.

AllowSoftPriceLimit Must be set to MMTS_TRUE.

[PositionType] Position type.

[Price] Price, FixReal.

[Yield] Yield, FixReal.

Quantity Quantity, a positive integer.

[VisibleQuantity] Visible quantity, a positive integer.

[BrokerRef] A character field which can be freely completed by the broker.

[ExpDate] MMTS date type.

[ExpTime] MMTS time type.

[TriggerPrice] Activation limit, FixReal.

[MinFillQty] Minimum quantity, a positive integer.

[TradeRef] Trade reference, for trade allocation.

[InternalRef] A character field which can be freely completed, it will not be sent to MMTS.

If the value of the given *optional* field is NOT_DEFINED or it is an empty string, then it will not be modified. Either the *Price* or the *Yield* field can be defined, but they are mutually exclusive. In the case of an order entry, the value of the *FixReal decimals* field will not be interpreted by MMTS. It only represents information for the client querying the order entry board (e.g. to how many decimals the number should be rounded off).

4. Values of the MMTS II order withdraw fields

The order entry record fields may receive the following values when an order is withdrawn:

[OrdNo] The order number of the order to be withdrawn. Its value can be found in the *OrdNo* field of the order record.

[**OrdNoSpeedIdx**] An internal index of the order in the MMTS trading system. (With the help of this, MMTS can find the requested order record more quickly.) Its value can be found in the *OrdNoSpeedIdx* field of the order record.

[**TrdAccId**] The character ID of the trading account.

[**BuySell**] BuySell type.

[**BoardId**] The character ID of the securities board. Its value can be found in the *BoardId* field of the security record.

[**InstrId**] The character ID of the instrument. Its value can be found in the *Id* field of the instrument record.

[**SecId**] The character ID of the security. Its value can be cut from the *Id* field of the security record.

[**Price**] Price.

[**Yield**] Yield.

[**BrokerRef**] A character field which can be freely completed by the broker.

OpCode Bool Operation, Boolop enumerated type.

PopCode Compare Operation, CompareOp. enumerated type

[**MultilegOrdNo**] The order number of the order to be modified. Its value can be found in the *MultilegOrdNo* field of the order record.

[**MultilegOrdNoSpeedidx**] An internal index of the order in the MMTS trading system. (With the help of this, the MMTS can find the requested order record more quickly.) Its value can be found in the *MultiLegSpeedIdx* field of the order record.

[**UserId**] User ID. Its value can be found in the *Id* field of the firm record.

[**FirmId**] Firm ID. Its value can be found in the *Id* field of the user record.

[**OrderDate**] MMTS Date type.

[**OrderTime**] MMTS Time type.

[**MarketMaker**] Market maker.

[**InternalRef**] A character field which can be freely completed, it will not be sent to the MMTS.

In the case of an `OpCode = BoolOp_And`, those orders will be withdrawn whose fields concurrently comply with the values given in the withdrawal (*and* relation). In the case of an `OpCode = BoolOp_Or`, it is enough if the fields of the order comply with only one of the values given during the withdrawal. At present only the `BoolOp_And` operator can be used in the MMTS. In the withdrawal, no fields with empty string or `NOT_DEFINED` values will be examined. When an order is entered, the value of the `FixReal decimals` field will not be interpreted by MMTS. It only represents information for the client querying the order entry board (e.g. to how many decimals the number should be rounded off).

ATTENTION !!!

If on deleting (withdrawing) orders the OrderNo field and the other non-mandatory (not obligatory) fields of the delete record (order cancellation record) are empty or 0, then all the orders belonging to the given firm

will be withdrawn. Therefore special care should be taken when the **OrderNo** field is left empty.

5. Values of the MMTS II Tick up/Tick Down fields

When initiating a Tick up/Tick down, the fields of the order entry record may accept the following values⁴:

[OrdNo] The order number of the order to be ticked up or down. Its value can be found in the *OrdNo* field of the order record.

[OrdNoSpeedIdx] An internal index of the order in the MMTS trading system. (With the help of this, MMTS can find the requested order record more quickly.) Its value can be found in the *OrdNoSpeedIdx* field of the order record.

[TrdAccId] The character string ID of a trading account.

[ExecutionId] The Id of the executor (user or algorithm)

[BuySell] BuySell enumerated type. For Tick up/Tick down it can specify both sides at the same time.

[BoardId] The character string ID of the trading board. Its value can be found in the *BoardId* field of the security record.

[SecId] The character string ID of the security. Its value may be cut from the *Id* field of the security record.

Tick Signed integer. (The number of ticks the price will be moved up or down with.)

[UserId] User ID. Its value can be found in the *Id* field of the user record.

[FirmId] Company ID. Its value can be found in the *Id* field of the firm record.

[BrokerRef] A character string field which can be freely completed by the broker. In this case (for tick up/tick down) it will not be sent to the MMTS.

[InternalRef] A character string field which can be freely completed, it will not be sent to the MMTS.

The price of those orders will be modified whose fields concurrently comply with the values given in the modification (tick up/tick down) order entry record (*and* relation). Empty fields or fields with a 0 value in the modification order entry record will not be taken into consideration.

⁴ The brackets imply non-mandatory data. If these are not given, then in the case of an INT type, we have to put in 0, the NOT_DEFINED value, while in the case of a STRING type, we have to put in spaces.

6. Values of the MMTS II Market Maker order fields

In case of a Market Maker order entry, the fields of the record specifying the market maker order may accept the following values:

BoardId The character string ID of the trading board. Its value can be found in the *BoardId* field of the security record.

SecId The character string ID of the security. Its value may be cut from the *Id* field of the security record.

OrderType An enumerated type. Limit or Market order is possible.

Duration An enumerated type for expiry.

PurgeOnLogoff Purge upon MMTS user logoff.

AllowSoftQtyLimit Must be set to MMTS_TRUE.

AllowSoftPriceLimit Must be set to MMTS_TRUE.

IsPrivate Private order. At present, it can only be an MMTS_FALSE.

[ExpDate] MMTS Date type.

[ExpTime] MMTS Time type.

BuySell BuySell enumerated type. For Market Maker Order entry it can specify both sides at the same time.

Replace In case of 1 existing Market Maker orders for the given security will be withdrawn, in case of 0, the withdrawal of the existing Market maker orders for the given security depends on the settings in the MMTS II.

[ExecutionId] The Id of the executor (user or algorithm)

[BuyTrdAccId] The character string ID of the trading account.

[BuyPrice] MMTS II fixreal type price.

[BuyYield] MMTS II fixreal type yield.

[BuyQuantity] Quantity, a positive integer.

[BuyVisibleQuantity] Quantity disclosed, a positive integer.

[BuyBrokerRef] A character string field for the Buy side order which can be completed by the broker freely.

[BuyPositionType] Enumerated type for the Buy side order position.

[BuyTradeRef] Trade reference for the Buy side order used for trade allocation.

[SellTrdAccId] The character string ID of the trading account.

[SellPrice] MMTS II fixreal type price.

[SellYield] MMTS II fixreal type yield.

[SellQuantity] Quantity, a positive integer.

[SellVisibleQuantity] Quantity disclosed, a positive integer.

[SellBrokerRef] A character string field for the Sell side order which can be completed by the broker freely.

[SellPositionType] Enumerated type for the Sell side order position.

[SellTradeRef] Trade reference for the Sell side order used for trade allocation.

[InternalRef] A character string field which can be freely completed, it will not be sent to MMTS.

7. Rules related to MMTS II Order Types

For certain order types the related order entry fields have to be also completed:

If for an order duration the Duration field specifies Duration_GoodTillTime then the **ExpTime** field has to be completed with a valid expiry time.

If for an order duration the Duration field specifies Duration_GoodTillDate then the **ExpDate** field has to be completed with a valid expiry date.

For Market Maker orders the following rules have to be observed when filling in the fields below:

OrderType Only OrderType_Limit is allowed.

Duration Only Duration_Session or Duration_Day is allowed, so a Market Maker order can have only a session or day expiry.

Notes
