

Trading Related Messages

- [OPEN_ORDERS_REQUEST](#)
- [ORDER_UPDATE](#)
- [OPEN_ORDERS_REJECT](#)
- [HISTORICAL_ORDER_FILLS_REQUEST](#)
- [HISTORICAL_ORDER_FILL_RESPONSE](#)
- [HISTORICAL_ORDER_FILLS_REJECT](#)
- [CURRENT_POSITIONS_REQUEST](#)
- [POSITION_UPDATE](#)
- [CURRENT_POSITIONS_REJECT](#)

OPEN_ORDERS_REQUEST [s_OpenOrdersRequest structure] Client >> Server

This is a message from the Client to the Server requesting the currently open orders.

The Server will send open/working orders in response to this request through [ORDER_UPDATE](#) messages.

The Server will not return canceled or filled orders.

When the Server responds to this request, it needs to respond with a separate [ORDER_UPDATE](#) for each order.

When the Server responds to this request, OrderUpdateReason in the [ORDER_UPDATE](#) message must be set to **OPEN_ORDERS_REQUEST_RESPONSE** indicating the orders are being restated.

If there are no Open orders, the Server will send back 1 [ORDER_UPDATE](#) message with only the TotalNumberMessages, MessageNumber, RequestID, OrderUpdateReason, NoOrders = 1 fields set in the [ORDER_UPDATE](#) message.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.

[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	A unique request identifier for this request.
[int32] RequestAllOrders	0 = request a specific order. 1 = for all orders (default).
[char] ServerOrderID	Leave this field empty if Client wants all orders. Otherwise, specify the specific ServerServerID that want an ORDER_UPDATE message for.
[char] TradeAccount	Leave this field empty if Client wants open orders for all Trade Accounts. Otherwise, specify the specific TradeAccount that want ORDER_UPDATE messages for.

ORDER_UPDATE [s_OrderUpdate structure] Server >> Client

The **ORDER_UPDATE** is a unified message from the Server to the Client which communicates the complete details of an order, the Order Status, and the reason for sending the message (OrderUpdateReason).

DTC uses this single unified message to provide an update for an order. The **OrderUpdateReason** field provides a clear indication for each reason this message is being sent.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automat

[unsigned int16] Type	The standard message type field. Automati
[int32] RequestID	<p>Set to 0 unless this is i</p> <p>OPEN ORDERS REQUEST, in which ca</p> <p>RequestID given in the OPEN ORDERS</p> <p>If this ORDER UPDATE is unsolicited, for</p> <p>other unsolicited order event, the Server m</p>
[int32] TotalNumMessages	<p>This indicates the total number of ORDER</p> <p>a batch of reports is being sen</p> <p>OPEN ORDERS REQUEST. If there is c</p> <p>this will be 1. The Server must use a va</p> <p>report. A Client should not rely on this field</p>
[int32] MessageNumber	<p>This indicates the 1-based index of the O</p> <p>when a batch of reports is being s</p> <p>OPEN ORDERS REQUEST. If there is c</p> <p>this will be 1. Use a value of 1 for an u</p> <p>should not rely on this field for an unsolici</p>
[char] Symbol	The symbol for the order.
[char] Exchange	The optional exchange for the symbol.

<p>[char] PreviousServerOrderID</p>	<p>Used upon a Cancel and (ORDER_CANCEL_REPLACE_COMPLE Order identifier is given.</p> <p>In this case this field needs to be set to identifier.</p> <p>This should be left at the default setting c the Server does not change the Serve Cancel and Replace operation.</p>
<p>[char] ServerOrderID</p>	<p>The ServerOrderID is set by the server order. When a new order is submitted by responds with an ORDER_UPDATE, this order identifier which is good for the life of</p> <p>This ServerOrderID can optionally change operation. In this case, the PreviousServe the previous ServerOrder ORDER_CANCEL_REPLACE_COMPLE</p> <p>ServerOrderID must always be set exce cases when the OrderUpdateReason NEW_ORDER_REJECTED, ORDI ORDER_CANCEL_REPLACE_REJECTE ClientOrderID must be set.</p>

<p><u>[char]</u> ClientOrderID</p>	<p>The ClientOrderID is the order identifier. When the Client submits a new order, existing order, or cancels an order, then this identifier.</p> <p>The Client must maintain the same order identifier for the duration of the order.</p> <p>The Server should persist the ClientOrderID. The <u>session</u> is defined as the period of time from the first connection between the Client and Server until the last connection.</p> <p>The Client should only rely upon the ClientOrderID when the OrderUpdateReason is one of the following: NEW_ORDER_ACCEPTED, NEW_ORDER_REJECTED, ORDER_CANCELED, ORDER_CANCEL_REJECTED, ORDER_CANCEL_REPLACE_REJECTED, ORDER_CANCEL_REPLACE_ACCEPTED.</p> <p>After a new order has been accepted, the Client should no longer rely upon the given ClientOrderID from the server to identify the order. The Client needs to maintain a copy of the ClientOrderID for order modifications and cancellations because the Server does not for those.</p>
<p><u>[char]</u> ExchangeOrderID</p>	<p>The order identifier from the exchange that is optional.</p>
	<p>This is required. Needs to be set to one of the following values to indicate the current status of the order:</p> <ul style="list-style-type: none"> • ORDER_STATUS_UNSPECIFIED: the order is unset. Use this when the Server does not know the order status when the Order Update is sent. • ORDER_STATUS_ORDER_SENDED: Client sends an order to the Server.

[OrderStatusEnum] OrderStatus

internally will set the ORDER_STATUS_ORDER_SE not set this Status.

- **ORDER_STATUS_PENDING_C** means the Server has accepted but not yet considered in a fully working reason.
- **ORDER_STATUS_PENDING_C** status applies to a Limit or Stop parent order. It will have this order has not yet filled.
- **ORDER_STATUS_OPEN (4):** working.
- **ORDER_STATUS_PENDING_C (5):** Order is pending a Cancel operation. The Server sends a [ORDER_UPDATE](#) message with OrderUpdateReason **GENERAL_ORDER_UPDATE** set to this status to indicate the Replace operation.
- **ORDER_STATUS_PENDING_C** is pending a Cancel operation. The Server sends a [ORDER_UPDATE](#) message with OrderUpdateReason **GENERAL_ORDER_UPDATE** set to this status to indicate the Cancel operation.
- **ORDER_STATUS_FILLED (7):** no longer working.
- **ORDER_STATUS_CANCELED** canceled. If the user tries to cancel an already been canceled, then the Server returns the Order Status for it.
- **ORDER_STATUS_REJECTED** rejected after the initial order is working.
- **ORDER_STATUS_PARTIALLY_FILLED** Order is partially filled and still working.

This is required. This field needs to be set to one of the values by the Server to indicate the reason for the order being sent.

- **OPEN_ORDERS_REQUEST_R**
When the [ORDER_UPDATE](#) is response to an [OPEN_C](#) request, this is the OrderUpdate
- **NEW_ORDER_ACCEPTED (2)**
indicates a new order has been
- **GENERAL_ORDER_UPDATE (**
update. For example, when
process of being canceled, the
[ORDER_UPDATE](#) message wi
set to **ORDER_STATUS_PEN**
this particular case the **OrderUp**
to be set to **GENERAL_ORDER**
- **ORDER_FILLED (4):** Upon a
order, this is the OrderUp
OrderUpdateReason must only
[ORDER_UPDATE](#) is sent at the
fill. A previously filled order that
response to an [OPEN_ORDEF](#)
not use this OrderUpdateReason
- **ORDER_FILLED_PARTIALLY (**
of the order, this is the Orderl
OrderUpdateReason must only
[ORDER_UPDATE](#) is sent at the
fill. A previously filled order that
response to an [OPEN_ORDEF](#)
not use this OrderUpdateReason
- **ORDER_CANCELED (6):** This (indicates the order is now succe
- **ORDER_CANCEL_REPLACE_ (**
This OrderUpdateReason indica
successfully canceled and repla
- **NEW_ORDER_REJECTED (8):**
been submitted by the Client, i
for any reason and was never
will send through an [ORDER](#)
OrderUpdateReason. In this cas
to set the **OrderStatus** in the [C](#)
ORDER_STATUS_REJECTED.

The following fields need
NEW_ORDER_REJECTED C
OrderUpdateReason,
MessageNumber,
ClientOrderID, Symbol, Exchang

- **ORDER_CANCEL_REJECTED**

[\[OrderUpdateReasonEnum\]](#) **OrderUpdateReason**

the Client to cancel the [CANCEL_ORDER](#) message has

The current status of the order **OrderStatus** member of the message.

In the case where the given [CANCEL_ORDER](#) message is known, then respond with a message and set the **OrderStatus** to **ORDER_CANCEL_REJECTED**.

In the case where the **ServerOrderID** is not set, then in the [ORDER_UPDATE](#) message, the **ClientOrderID** must always be set in the [CANCEL_ORDER](#) message.

In the case where the order is canceled for the given [CANCEL_ORDER](#) message, then set the **OrderUpdateReason** to **ORDER_CANCEL_REJECTED**.

If for some reason the Server is not aware of the status of the order that was canceled, then set the **OrderStatus** to **ORDER_STATUS_UNSPECIFIED**.

- **ORDER_CANCEL_REPLACE** request by the Client to cancel a message with the [CANCEL_REPLACE](#) has been rejected.

The current status of the order **OrderStatus** member of the message.

In the case where the given [CANCEL_REPLACE_ORDER](#)

	<p>Client is not known, then ORDER_UPDATE message</p> <p>OrderUpdateReason</p> <p>ORDER_CANCEL_REPLACE_ the OrderStatus to ORDER_ST</p> <p>In the case where the ServerOr then in the ORDER_UPDATE not be set. However, in this cas set the ClientOrderID in th message to the given Clic CANCEL_REPLACE_ORDER ClientOrderID must always be s ORDER_UPDATE.</p> <p>If for some reason the Server is status of the order that was canceled and replaced, then set ORDER_STATUS_UNSPECIFII</p>
[OrderTypeEnum] OrderType	<p>The order type. Can be set to one of the fo</p> <ul style="list-style-type: none"> • ORDER_TYPE_MARKET • ORDER_TYPE_LIMIT • ORDER_TYPE_STOP • ORDER_TYPE_STOP_LIMIT • ORDER_TYPE_MARKET_IF_T
[BuySellEnum] BuySell	<p>Indicates whether the order is a Buy or Se of the following constants: DTC::BUY (1) o</p>
[double] Price1	<p>For orders that require a price, this is the c</p> <p>For binary encoding, if this field is not DBL_MAX. Refer to Unset Message Fields</p>

[double] Price2	<p>For Stop-Limit orders this is the Limit price.</p> <p>For binary encoding, if this field is not DBL_MAX. Refer to Unset Message Fields.</p>
[TimeInForceEnum] TimeInForce	<p>The Time in Force of the order. Can be any of the following:</p> <ul style="list-style-type: none"> • TIF_DAY • TIF_GOOD_TILL_CANCELED • TIF_GOOD_TILL_DATE_TIME • TIF_IMMEDIATE_OR_CANCEL • TIF_ALL_OR_NONE • TIF_FILL_OR_KILL
[t_DateTime] GoodTillDateTime	<p>The expiration Date and Time of the TimeInForce is TIF_GOOD_TILL_DATE_TIME.</p>
[double] OrderQuantity	<p>The quantity of the order.</p> <p>For binary encoding, if this field is not DBL_MAX. Refer to Unset Message Fields.</p>
[double] FilledQuantity	<p>The number of shares or contracts that have been filled.</p> <p>For binary encoding, if this field is not DBL_MAX. Refer to Unset Message Fields.</p>
[double] RemainingQuantity	<p>The number of shares or contracts that still remain unfilled.</p> <p>For binary encoding, if this field is not DBL_MAX. Refer to Unset Message Fields.</p>

[double] AverageFillPrice	<p>The average price of all of the fills for the c</p> <p>For binary encoding, if this field is not DBL_MAX. Refer to Unset Message Fields</p>
[double] LastFillPrice	<p>The price of the most recent fill.</p> <p>Only necessary to set if OrderUpdateRea ORDER_FILLED_PARTIALLY.</p> <p>For binary encoding, if this field is not DBL_MAX. Refer to Unset Message Fields</p>
[t_DateTime] LastFillDateTime	<p>The date and Time of the most recent fill.</p> <p>Only necessary to set if OrderUpdateRea ORDER_FILLED_PARTIALLY.</p>
[double] LastFillQuantity	<p>The number of contracts/shares that has fill that is currently reported through the OF</p> <p>Only necessary to set if OrderUpdateRea ORDER_FILLED_PARTIALLY.</p> <p>For binary encoding, if this field is not DBL_MAX. Refer to Unset Message Fields</p>
[char] LastFillExecutionID	<p>The unique identifier for the most recent fill</p> <p>Only necessary to set if OrderUpdateRea ORDER_FILLED_PARTIALLY.</p>
[char] TradeAccount	<p>The trade account the order belongs to.</p>

[char] InfoText	Free-form text with information to compr When an order is rejected, this should indicate the reason for the rejection.
[unsigned int8] NoOrders	Set by the Server to 1 to indicate th OPEN ORDERS REQUEST message b being responded to. Otherwise, leave at th
[char] ParentServerOrderID	This is the ServerOrderID of the parent ord Order Update Report is for, is a child Otherwise, this is an empty text string.
[char] OCOLinkedOrderServerOrderID	In the case of an OCO orde SUBMIT NEW OCO ORDER , whether it this is the ServerOrderID of the other ord two orders are considered "linked" toget empty text string.
[OpenCloseTradeEnum] OpenOrClose	For the description for this field, refer to O
[char] PreviousClientOrderID	The PreviousClientOrderID is the previo by the Client for the order, if the Client cha and replace request or an order cancel rec A Server only is obligated to provide this ClientOrderID has been changed. Subse messages do not need to set this field.
[char] FreeFormText	This is the optional free-form text that wa order using the new order messages .

[t_DateTime] OrderReceivedDateTime	This is the Date-Time when the original Server. Order modifications normally will r be updated.
--	---

OPEN_ORDERS_REJECT [[s_OpenOrdersReject structure](#)] Server >> Client

If the Server is unable to serve the request for an [OPEN_ORDERS_REQUEST](#) message received, for a reason other than there not being any open orders, then send this message to the Client.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	This is set to the RequestID field sent in the OPEN_ORDERS_REQUEST message from the Client.
[char] RejectText	Free-form text indicating the reason for rejection.

HISTORICAL_ORDER_FILLS_REQUEST [[s_HistoricalOrderFillsRequest structure](#)] Client >> Server

This is a message from the Client to the Server to request order fills/executions for an order or orders.

The Server will respond with one or more [HISTORICAL_ORDER_FILL_RESPONSE](#) messages or a single [HISTORICAL_ORDER_FILLS_REJECT](#) message.

--

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	A unique request identifier. The Server will return the same identifier in the response.
[char] ServerOrderID	Leave empty if want all order fills. Otherwise, request order fills for given Server Order identifier.
[char] TradeAccount	<p>This field specifies the particular Trade Account to request order fills for.</p> <p>This field is normally required to be set. Some Servers may support an empty field in which case the Server will respond with the historical order fills for all trade accounts the logged in username has access to. If a Server does support responding with historical order fills for all trade accounts, then it will send a <u>separate response</u> for each trade account.</p>
[int32] NumberOfDays	<p>The NumberOfDays field is ignored by the Server when StartDateTime is set.</p> <p>The NumberOfDays field specifies to the Server to return order fills counting from the current day back by the specified number of days.</p> <p>If NumberOfDays and StartDateTime are both not set or 0, the Server will return all historical order fills available.</p>

[t_DateTime] StartDateTime	<p>The NumberOfDays field is ignored by the Server when StartDateTime is set.</p> <p>The StartDateTime field specifies to the Server to return order fills starting with date time specified.</p> <p>If NumberOfDays and StartDateTime are both not set or 0, the Server will return all historical order fills available.</p>
--	---

HISTORICAL_ORDER_FILL_RESPONSE

[s_HistoricalOrderFillResponse structure] Server >> Client

This is a message from the Server to the Client providing an individual historical order fill in response to a [HISTORICAL_ORDER_FILLS_REQUEST](#) message.

The Server is expected to send this message to the Client in response to a [HISTORICAL_ORDER_FILLS_REQUEST](#) message even when there are no order fills to return. If there are no order fills to return, it needs to set the **NoOrderFills** field to 1.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	The RequestID specified in the HISTORICAL_ORDER_FILLS_REQUEST message from the Client.
[int32] TotalNumberMessages	This indicates the total number of order fill reports when a batch of reports is being sent. If there is only one order fill report being sent, this will be 1.

[int32] MessageNumber	This indicates the 1-based index of the order fill report when a batch of reports is being sent. If there is only one order fill report being sent, this will be 1.
[char] Symbol	The symbol the order fill is for.
[char] Exchange	The optional exchange for the symbol.
[char] ServerOrderID	This is the Server provided order identifier that the order fill was associated with.
[BuySellEnum] BuySell	This is the fill side and can be either BUY or SELL.
[double] Price	This is the price of the order fill.
[t_DateTime] DateTime	This is the Date and Time of the order fill.
[double] Quantity	This is the quantity of the order fill.
[char] UniqueExecutionID	This is the unique execution identifier for the order fill.
[char] TradeAccount	This is the trade account that the order fill is associated with.

[OpenCloseTradeEnum] OpenClose	<p>Indicates whether this is an opening or closing order fill.</p> <p>This field is optional.</p>
[unsigned int8] NoOrderFills	<p>Set to a numeric 1 to indicate there are no historical order fills.</p> <p>If there are no order fills to return, the Server needs to set this to 1 and send through 1 HISTORICAL_ORDER_FILL_RESPONSE message to indicate there are no order fills. Otherwise, leave this field at the default of 0.</p>

HISTORICAL_ORDER_FILLS_REJECT [s_HistoricalOrderFillsReject structure] Server >> Client

If the Server is unable to serve the request for a [HISTORICAL_ORDER_FILLS_REQUEST](#) message received, for a reason other than there not being any historical order fills, then send this message to the Client.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	This is set to the RequestID field sent in the HISTORICAL_ORDER_FILLS_REQUEST message.

[char] RejectText	Free-form text indicating the reason for rejection.
---	---

CURRENT_POSITIONS_REQUEST [s_CurrentPositionsRequest structure] Client >> Server

This is a message from the Client to the Server to request the current open Trade Positions.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by constructor.
[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	The request identifier. The Server will send this back in the POSITION_UPDATE response messages.
[char] TradeAccount	This is an optional field. Leave this empty to request the Server to return all current open Trade Positions for all trade accounts on the logged in Username. Otherwise, specify a particular trade account to request Trade Positions for.

POSITION_UPDATE [s_PositionUpdate structure] Server >> Client

This is a message from the Server to the Client to report a Trade Position for a symbol in any Trade Account for the logged in Username.

The Position Update message can either be solicited, in response to [CURRENT_POSITIONS_REQUEST](#). Or unsolicited as a Trade Position for a symbol changes during the connection to the Server. Each Trade Position is contained within a single message.

When the server is responding with one or more **POSITION_UPDATE** messages in response to a [CURRENT_POSITIONS_REQUEST](#) message, it must not send any unsolicited **POSITION_UPDATE** messages interleaved with the solicited **POSITION_UPDATE** messages in

response to the [CURRENT_POSITIONS_REQUEST](#) message.

Field Name	Field Description
[unsigned int16] Size	The standard message size field. Automatically set by
[unsigned int16] Type	The standard message type field. Automatically set by
[int32] RequestID	<p>The Server sets this to 0, the default, if this is a update.</p> <p>Otherwise, when the Server is sending Trade Po CURRENT_POSITIONS_REQUEST message, it RequestID given in the CURRENT_POSITIONS_RE</p>
[int32] TotalNumberMessages	<p>This indicates the total number of Position Update m messages is being sent. If there is only one Positio sent, this will be 1. Use a value of 1 for an unsolicited</p> <p>A Client should not rely on this field for an unsc required to be set.</p>
[int32] MessageNumber	<p>This indicates the 1-based index of the Position Upda of messages is being sent. If there is only one Positic sent, this will be 1. Use a value of 1 for an unsolicited</p> <p>A Client should not rely on this field for an unsc required to be set.</p>
[char] Symbol	The symbol for the Position.

[char] Exchange	The optional exchange for the symbol.
[double] Quantity	The quantity of the current Position for the symbol. A long Position and a negative number is for a short Position.
[double] AveragePrice	The average position price.
[char] PositionIdentifier	<p>When the <code>Server::LOGON_RESPONSE::UsesMultiplePositionsPerSymbol</code> is set to 1, then it must set PositionIdentifier to a unique identifier for the particular Symbol and Trade Account.</p> <p>Otherwise, this field is optional and it is recommended if it does not provide multiple positions for a Symbol and Trade Account.</p> <p>This field identifies an individual Trade Position in the multiple Positions for a particular Symbol and Trade Account.</p> <p>A Client must implement support for <code>Server::LOGON_RESPONSE::UsesMultiplePositionsPerSymbol</code> set to 1.</p>
[char] TradeAccount	The trade account the Trade Position is in for the Symbol.

[double] HighPriceDuringPosition	<p>HighPriceDuringPosition is the highest price the symbol has reached during the life of the Trade Position. This value can be reset by a change in Trade Position Quantity.</p> <p>This is an <u>optional</u> field for the Server to provide.</p>
[double] LowPriceDuringPosition	<p>LowPriceDuringPosition is the lowest price the symbol has reached during the life of the Trade Position. This value can be reset by a change in Trade Position Quantity.</p> <p>This is an <u>optional</u> field for the Server to provide.</p>
[double] QuantityLimit	<p>This is the limit or maximum Trade Position Quantity for the Trade Account and Symbol. This applies to both long and short positions.</p> <p>This is only an informational value provided from the Server.</p> <p>This is an <u>optional</u> field for the Server to provide.</p>

CURRENT_POSITIONS_REJECT [s_CurrentPositionsReject structure] Server >> Client

If the Server is unable to serve the request for an [CURRENT_POSITIONS_REQUEST](#) message received, for a reason other than there not being any current Trade positions, then send this message to the Client.

This must never be sent when there are actually no Trade Positions in the account or accounts requested.

Field Name	Field Description
[unsigned int16] Size	<p>The standard message size field. Automatically set by the constructor.</p>

[unsigned int16] Type	The standard message type field. Automatically set by constructor.
[int32] RequestID	This is set to the RequestID field sent in the CURRENT POSITIONS REQUEST message.
[char] RejectText	Free-form text indicating the reason for the rejection.

*Last modified Monday, 16th January, 2023.