

## GetCommissions – always returns 0

Via the FireOrderChanged the `serializeOrderChange` Method is invoked

```
public void onMessage(IMessage message) throws JFException
```

```
{
    _log.debug(message.toString());

    switch (message.getType())
    {
        case ORDER_SUBMIT_OK:
        case ORDER_SUBMIT_REJECTED:
        case ORDER_FILL_OK:
        case ORDER_FILL_REJECTED:
        case ORDER_CLOSE_OK:
        case ORDER_CLOSE_REJECTED:
        case ORDER_CHANGED_OK:
        case ORDER_CHANGED_REJECTED:
            IOrder order = message.getOrder();
            if (order != null) {
                fireOrderChanged(order);
            } else {
                _log.debug("Order is null.");
            }
            break;
        default:
            break;
    }

    processPendingMessages();
}
```

```
private String serializeOrderChange(IOrder order) throws Exception
```

```
{
    State state = order.getState();
    double filledAmount = state == State.FILLED || state == State.CLOSED ? order.getAmount() : 0;

    StringBuilder sb = new StringBuilder("OrderChange;");
    sb.append(order.getId());
    sb.append(";");
    sb.append(order.getLabel());
    sb.append(";");
    sb.append(order.getInstrument());
    sb.append(";");
    sb.append(order.getOrderCommand().toString());
    //sb.append(Order.getOrderSide(order.getOrderCommand()).toString());
    //sb.append(";");
    //sb.append(Order.getOrderType(order.getOrderCommand()).toString());
    sb.append(";");
    sb.append((int)normalizeQuantity(order.getRequestedAmount()));
    sb.append(";");
    sb.append((int)normalizeQuantity(filledAmount));
}
```

```
sb.append(";");
sb.append(Convert.toString(order.getOpenPrice()));
sb.append(";");
sb.append(state.toString());
sb.append(";");
switch (order.getState()) {
    case CLOSED:
        sb.append(order.getCloseTime());
        sb.append(";");
        break;
    case FILLED:
        sb.append(order.getFillTime());
        sb.append(";");
        break;
    default:
        sb.append("0;");
}
sb.append(order.getGoodTillTime());
sb.append(";");
sb.append(Convert.toString(order.getCommission())); // always returns 0,00

return sb.toString();
}
```

# GetOrders doesn't deliver what we expect.

We have the following code to request orders information:

```
List<Order> orders = _strategy.getOrders();
for (Order order : orders) {
    _log.debug(order.getId() + "\t" + order.getLabel() + "\t" + order.getInstrument());
}
```

Where `_strategy.getOrders` is implemented in the following way

```
public List<Order> getOrders() throws JFException
{
    IEngine engine = _context.getEngine();
    if (engine == null)
        return null;

    return engine.getOrders();
}
```

In DukasCopy application we have 2 opened positions and 1 order

The screenshot displays the DukasCopy application interface with the following sections:

- AUD/USD Order Entry:** Shows a bid of 1.03978 and an ask of 1.03991. Buttons for 'Place Bid' and 'Place Offer' are visible.
- Market Depth:** A table showing bid and ask orders with volumes.
- Instruments:** A list of various currency pairs like EUR/AUD, GBP/CHF, etc.
- Position Summary:** A table showing two open positions:
 

| Instrument | Direction | Long/Short              | Amount    | Price  | P/L       |
|------------|-----------|-------------------------|-----------|--------|-----------|
| GBP/CHF    | LONG      | 0.001 ml. / 0 ml. (1/0) | GBP 0.001 | 1.5028 | EUR 0.52  |
| AUD/USD    | LONG      | 0.004 ml. / 0 ml. (2/0) | AUD 0.004 | 1.0406 | EUR -2.53 |
- Position Structure:** A table showing the details of the two positions, including their IDs and current values.
- Orders:** A table showing one pending order:
 

| Time               | Pos. ID                 | Order ID  | Instrument | Side | Amount    | Order | Price         | Type | Status  | Expl... |
|--------------------|-------------------------|-----------|------------|------|-----------|-------|---------------|------|---------|---------|
| 2012-11-01 14:2... | [Fillvmb] 273958AUD/USD | 168761397 | AUD/USD    | BUY  | AUD 0.001 | ENTRY | ASK ≥ 1.04147 | STOP | PEND... | GTC     |
- Messages:** A log of system messages, including order acceptance, filling, and cancellation.

At the bottom, the status bar shows: Equity: EUR 205.75 | Free Trading Line: EUR 16,116.00 | Profit/Loss: EUR -2.01 | Use of Leverage: 22% | One Click | Connected | Detached: 0

So we expect that getOrder function will return 3 items.

But in the log we have just 2 entries:

```
2012-11-01 18:16:54.041 DEBUG RequestHandler - 168761397 jf1Inximb AUD/USD
2012-11-01 18:16:54.041 DEBUG RequestHandler - 273958AUD/USD 273958AUD/USD
AUD/USD
```

1 item is for position and 1 for order.

Let's reconnect, cancel order and call this function again. In the result we have empty collection.

We expected 2 entries (for positions).

After this we can reconnect and try again. In this case we have 2 entries in the log:

```
2012-11-01 18:27:29.968 DEBUG RequestHandler - 273958AUD/USD 273958AUD/USD
AUD/USD
2012-11-01 18:27:29.968 DEBUG RequestHandler - 273958GBP/CHF 273958GBP/CHF
GBP/CHF
```

Both are for positions – exactly what we expected.