Order Management System

iTrader Server Interface Specification

Version 1.18.6.6

Last Updated: Oct 12, 2018

Table of Contents

1. Introduction 4
1.1 Document Notation 4
2 System Overview 5
2.1 Message Format 5
2.2 iTrader Server 6
2.3 Initial Workflow 7
3 User Authentication 8
3.1 System Users 8
3.2 Creating Login Session 8
3.3 Verify Session 11
3.4 Session Verification on iTS 12
3.5 Terminating Session 13
3.6 SSM Error Messages 14
3.7 Checking Password 16
3.8 Changing Password 17
3.9 Changing PIN 19
3.10 Verifying PIN 20
4 iTrader Server Operation 22
4.1 Overview 22
4.2 Session Maintenance 23
4.3 General Account Information 24
4.4 Account Position 26
4.5 Order History 29
4.6 Custom Information Query 31
4.7 Suborder Query 32
4.8 Trade Query 33
4.9 Money Voucher Query 34
4.10 ITS Error Message 35
5 Order Interface 37
5.1 Overview 37

5.2 Order Book 38

5.3 Placing Order 40

5.4 Calculate Fee & Charges 43

5.5 Changing Order 45

5.6 Cancelling Order 46

5.7 Error Handling 47

Appendix 1 – List of Attributes 48

Appendix 2A – Order Management during Non-Continuous Session 51

Appendix 2B – Order Behavior during Pre-open Session 52

Appendix 4 – List of Error Code 53

Appendix 5 – Order State Change Matrices 56

1. Introduction

This document specifies the details of communication protocol between Order Management System (OMS) and remote client applications. It defines the message format and procedure on different operation a client application can perform on OMS. It also provides client application developer a general concept on the system architecture.

The intended reader of this document is the technical personnel of the application development party. It is assumed that the reader has sufficient business knowledge on stock market and trading system in Hong Kong. This document does not cover a full specification on each server component but sufficient information for client application developers to develop their own system cooperating with OMS.

1.1 Document Notation

The meaning for message element format are listed below:

- Char: An ASCII string with unlimited length. Any character apart from pipe (0x7C), return (0x0D), linefeed(0x0A) and null(0x00) is allowed.
- Char(*n*): An ASCII string with maximum length of *n* bytes.
- Integer: A string of character “0” to “9”. Sign symbol (+/-) is allowed. Integer value should be ranged within 2^31-1.
- Integer(*n*): A string of character “0” to “9” with maximum length of *n.* Sign symbol (+/-) is allowed.
- Float: A string of character “0” to “9” with unlimited length. A decimal point “.” is allowed to make the string being interpret as a float point number. Sign symbol(+/-) is allowed.
- Date: A string of standard date format “yyyy/mm/dd”.
- Time: A string of standard time format “hh:mm:ss”.

2 System Overview

2.1 Message Format

All servers accept TCP connection and work upon valid message from client is received. A valid message is a string of ASCII text arranged in the following format:

```
<command|>[arg1|arg2|…|][attr1|value1|attr2|value2|…]
```

Elements, such as command or arguments, in the message are separated by the pipe character "|". So, all the elements should not contain such character unless it is a tailing byte of a two-byte character. Messages can be transmitted with or without encryption through TCP socket. If a stream of messages is transmitted to the server without encryption, they should be delimited by a return character plus an optional linefeed character. The server will not respond until it received a full message.

Generally, a message consists of a message command, command arguments and a list of attribute-value pair. The number of arguments depends on the corresponding command. Here is a sample message to iTrader Server:

```
ITS|verify|0|09348793874783|13|userid@127.0.0.1|
```

The above message consists of six elements. The first element "ITS" is a system directive for iTrader Server. The second element "verify" is the instruction to corresponding system. All commands are case insensitive unless stated otherwise. The other elements are the list of attribute-value pair. A valid attribute-value pair is consists of an attribute ID and an attribute value. An attribute ID must be an unsigned integer in string format. It is used to indicate the meaning of following attribute value. In the above example, the attribute "13" represents the user identity. The format of attribute values depends on specific attribute. Virtually, there is no restriction on the length of each element.

Some attributes for a command can be optional. Attributes can be arranged in any order within the message. Note also, that occasionally message might be extended by new attributes. The client should expect to see and ignore unknown attributes.

2.2 iTrader Server

iTrader Server (iTS) is an open order interface encapsulating order message flow between client application and different server component, integrated with security and user authentication checking. It provides client application which running on an open network, such as Internet, a standard and unified channel for access of OMS. Below is a diagram showing the role iTS playing:

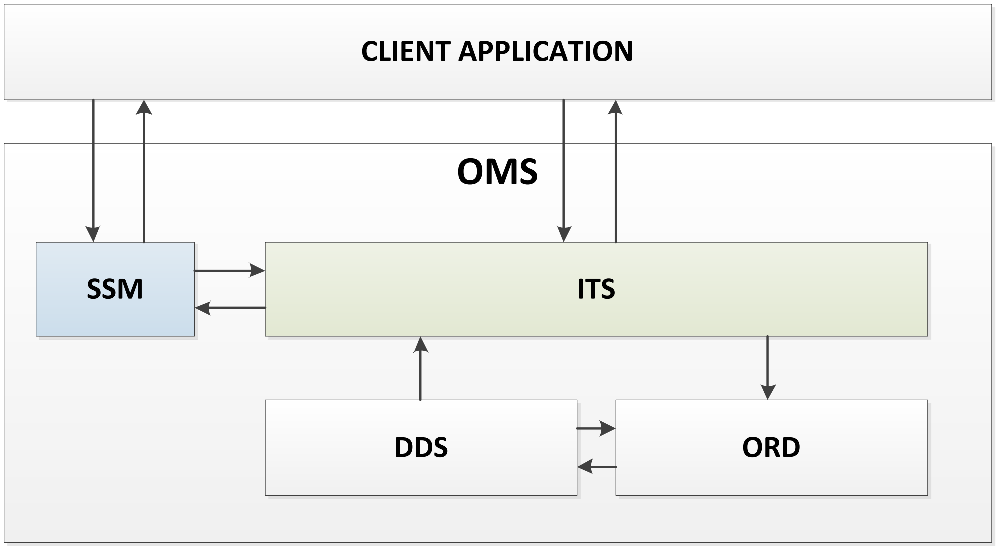

Diagram 2.3.1 – Server components relation diagram

After client application created a login session on SSM, it will gain access to iTS. It provides connecting clients all an order management application needs by routing suitable instruction to ORD and DDS. All feedback from ORD and DDS will be filtered and captured by iTS and then distribute to corresponding client. So, all order related operation can be done by a single socket connection.

2.3 Initial Workflow

Before working with iTS, the logon procedure must be done to SSM. After logon successfully, client application will get a session ID, the system token throughout the whole trading life cycle, and client application should make use of this session ID to verify iTS to finish the system authentication.

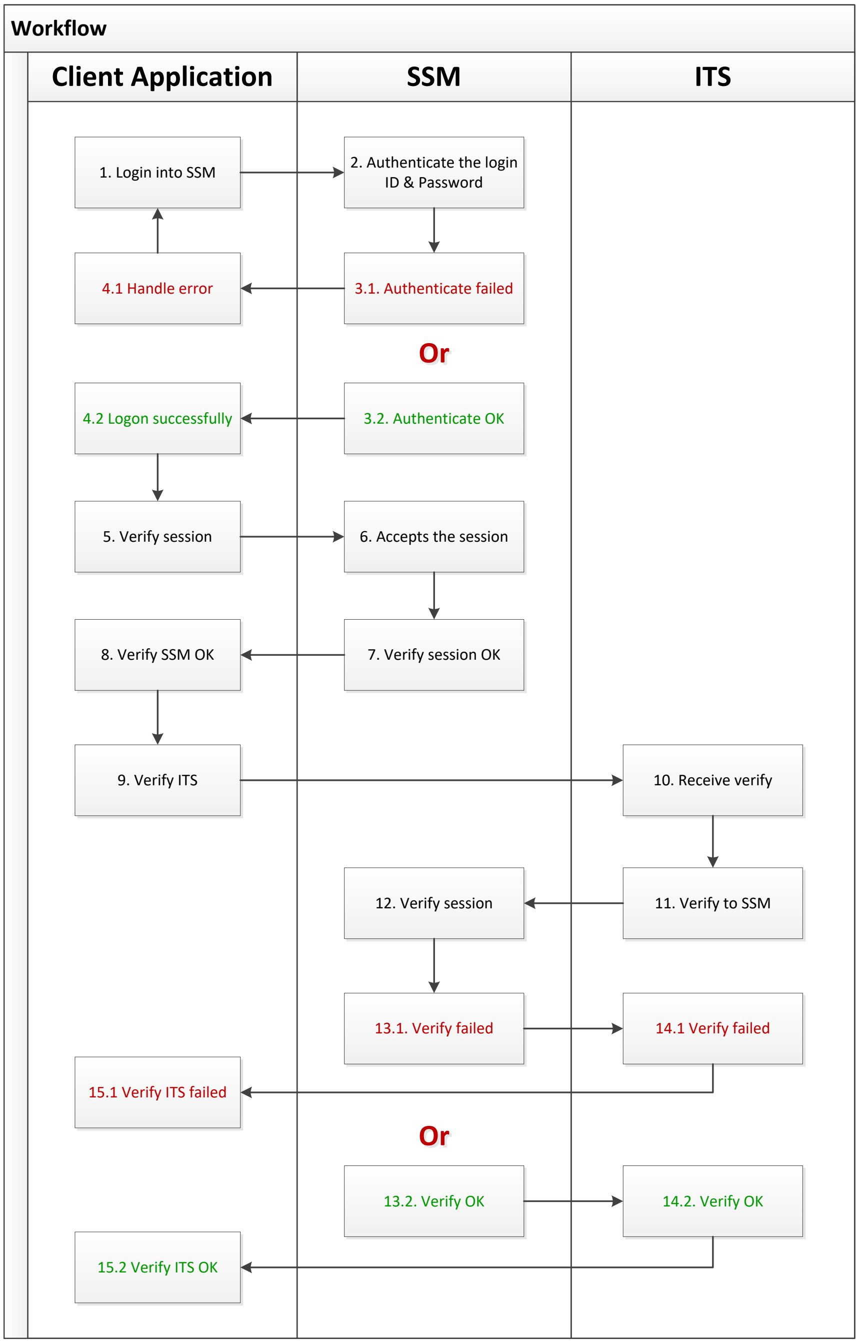

Diagram 2.4.1 – iTS logon workflow

3 User Authentication

3.1 System Users

Every user must have a user account to access the OMS. Client application has to create a login session before a user can access the OMS. The session must be maintained as long as the user accessing the system. Every user account can have only one login session at any particular moment. If a user attempt to create another session, the older session will be terminated immediately.

3.2 Creating Login Session

To create a login session, client application should first open a TCP socket and send a login message to SSM. Here is the format of the login message:

```
login|13|<user identity>|21|<password>|22|<account type>|157|<device>|99|<app ID>|115|<client IP>|
```

Example:

```
login|13|CC0001|21|1234|22|1|157|Internet|99|xTrader|115|123.123.123.123|
```

Table 3.2.1 – SSM login attribute summary

Name	ID	Format	Meaning
OMS_USER_NO	13	Char	Account ID of the user. Case insensitive.
OMS_NAME	21	Char	Account password. Case sensitive.
OMS_PRODTYPE	22	Char	Account type: 0=AE, 1=account (by default)
OMS_DEVICENAME	157	Char	Name of device
OMS_MACHINE_ID	520	Char	Optional. Used for device with serial number.
OMS_APP_ID	99	Char	Application ID for licensing.
OMS_APP_VER	110	Char	Application version
CLIENT_IP	115	Char	IP address of the client

A user may login to different device by different password. The name of device specifies the trading device the user trying to login. For Internet trading, simply specify “Internet” as attribute value for the name of device.

If user logon the system successfully, SSM will reply a message with session ID and user identity to the client application in the following format:

session|0|<session id>|13|<user identity>|22|<user type>|…

Example:
session|0|0105225441012552221634604247707733337551022834821263072598144511|13|CC0001@203.1.12.123|22|1|200|SEHK|201|HKFE|

Table 3.2.2 – SSM login response attribute summary

Name	ID	Format	Meaning
OMS_SYMBOL	0	Integer(64)	Unique session ID
OMS_USER_NO	13	Char	User identity, formatted as <userid>@<host>
OMS_PRODTYPE	22	Char	Account type
OMS_MATURITY	29	DateTime	Password expiry date time, in format "yyyy/MM/dd HH:mm:ss"
OMS_PIN_POS	116	Char	Random PIN number positions (optional), separated by comma
OMS_TRADE_DATE	134	DateTime	Server side date time, in format "yyyy/MM/dd HH:mm:ss"
OMS_FIRSTROW	200-299	Char	User Entitlement (optional)
OMS_LAST_EXETIME	1115	DateTime	Last logon time, in format "yyyy-MM-dd HH:mm:ss"

Client application has to keep the returned session ID and user identity in order to provide other server components for user verification. User entitlement is a list of string indicating whether a user is entitled to do a particular operation. The detail of entitlement mechanism is out of the scope of this document.

The PIN number positions are optional for those needs PIN number required, and SSM will generate random positions of the PIN number to require the authenticated end user to re-submit the PIN number in the specified positions when verifying the PIN number.

3.3 Verify Session

The major operation of SSM is to provide other server component a way to verify whether a login session of a particular user remains active, as well as checking user entitlement. Another usage of session verification, which is an important duty of a client application, is to keep the login session active. A login session will expire if there is no verification operation on certain period of time (default is 15 minutes). The timer will be renewed if a verification message of that particular session is received before it is expired. Here is the format of a verification message:

```
verify|0|<session id>|13|<user identity>|
```

Example:
verify|0|0105225441012552221634604247707733337551022834821263072598144511|13
|**CC0001@203.1.12.123**|

Table 3.3.1 – SSM verification attribute summary

Name	ID	Format	Meaning
OMS_SYMBOL	0	Integer(64)	Unique session ID
OMS_USER_NO	13	Char	User identity, formatted as <userid>@<host>

If the login session is verified successfully, SSM will reply client application with a session message, where the message format is exactly the same as the response of login operation. Please refer to Table 3.2.2 for detail.

Since SSM do not save any state information for any socket client connection, client application is not required to keep the socket connection to SSM after it finished its operation unless it need to receive notification of session termination, and this will be explained in later section.

3.4 Session Verification on iTS

When the client application is maintaining a login session for a user, it can open a TCP socket to iTS and start verifying login session. The purpose of verifying session on iTS is to bind socket connection with a particular login session. iTS will not accept any instruction before a login session is verified successfully. Here is the format of the message required to verify login session:

```
ITS|verify|0|<session id>|13|<user identity>|
```

Example:

```
ITS|verify|0|0105225441012552221634604247707733337551022834821263072598144511
|13|CC0001@203.1.12.123|
```

Table 3.4.1 – iTS verification attribute summary

Name	ID	Format	Meaning
OMS_SYMBOL	0	Integer(64)	Unique session ID
OMS_USER_NO	13	Char	User identity, formatted as <userid>@<host>

After iTS received a verification request, it will verify the given session with SSM. If the login session verified successfully, access of corresponding user account will be granted. Response from SSM will be route directly to client application no matter session verified successful or not. Please refer to previous session for message format.

3.5 Terminating Session

Login session will be terminated in one of the following cases:

1. Logout by client application.
2. Login session expired.
3. Being kicked by system operator.
4. User attempt to create another login session.

Client application should logout the session when it is no longer needed. To logout a session, a logout message in the below format should be sent to SSM.

logout|0|<session id>|13|<user identity>|

Example:

logout|0|0105225441012552221634604247707733337551022834821263072598144511|13
|**CC0001@203.1.12.123**|

Table 3.5.1 – SSM session logout attribute summary

Name	ID	Format	Meaning
OMS_SYMBOL	0	Integer(64)	Unique session ID
OMS_USER_NO	13	Char	User identity, formatted as <userid>@<host>

It will echo the message if the session terminated normally.

When a login session is terminated, a notification message will be sent explicitly to all connecting socket client that has verified the terminated session. This notifies the applications to terminate any operation for that particular user. The notification message will be in the following format:

disconnect|0|<session id>|13|<user identity>|

Example:

disconnect|0|010522544101255222163460424770773333755102283482126307259814451
1|13|**CC0001@203.1.12.123**|

Table 3.5.2 – SSM session terminate notification attribute summary

Name	ID	Format	Meaning
OMS_SYMBOL	0	Integer(64)	Unique session ID
OMS_USER_NO	13	Char	User identity, formatted as <userid>@<host>

3.6 SSM Error Messages

Error message will be sent to client application when the instruction is invalid or requesting operation executed failed. All error messages should be formatted as follow:

```
Error|25|<error message>|39|<error code>|
```

Example: ERROR|25|Login Failed|39|10|

Table 3.6.1 – Defined SSM error message summary

Code	Message	Possible Reason
0	No error	Password checking successfully
10	Login failed	User/account not found, or incorrect password
11	Password change failed	Old password incorrect, or user/account not found
12	Password Expire	Account disable or password expired. Password change is required if it’s expired
13	User suspended	The account is currently suspended
14	Password checking failed	The password for the device is invalid
17	Password change failed	New password conflict with existing old password
18	PIN change failed	New PIN number conflict with existing old PIN number
19	PIN Expired	PIN number has expired
20	Verify failed	Verifying session not found

Code	Message	Possible Reason
21	PIN change failed	Old PIN number incorrect
30	Remote verify failed	Host specified by user identity connection failed
40	Logout failed	Specified login session not found
41	Operation denied	Connected client do not have enough permission
42	TokenType error	Token type not in -1,0,1,2,3,4,5
43	It has error about OTP	OTP error
44	resend time is not ready yet	Request OTP before resend time ready
45	OTP time out	OTP has timeout
46	It has activated	Activate the activation code which has activated
47	it has no activate info	The id has no record in TokenTable
48	Activation code time out	Activation code has time out
50	Session Terminated due to new session	The same account repeat login
51	Session Expired	Session Expired
52	Session is kicked	kill session
53	User logout normally	User logout normally
54	Session Terminated due to verify retry failed	Session Terminated due to verify retry failed
55	activation code error	Activate code not in TokenTable
60	No certificate of <APP ID>	APP ID not defined in the cert
61	License exceeds limit of <APP ID>.<Online Count>/<Total Count>	Current APP ID logon session exceed the limitation defined in the cert
69	It can not duplicate Login in cert	User duplicate login when Cert can not duplicate

Code	Message	Possible Reason
70	PIN number verify failed	PIN number incorrect, or user not found
71	Challenge not found	Challenge session ID not found
72	Login denied	Login session denied by end user
73	Challenge timeout	Challenge session timeout
74	Challenge invalid	Challenge session invalid
76	notify address not exist	Can not find the notify address from database
80	Verify token failed	Token not match
81	Update database failed	Update database failed
82	Session ID failed	Session ID error
83	Token verification already done.	Token verify already done
84	Token verification required.	Verify token request
85	Service not available	Service not available
86	Send notify failed	Send notify message failed or timeout
87	User or Account not found in DB	Login ID does not exist on database
88	Resend times exceed resend limit	Resend OTP times exceed resend limit
89	Insufficient token key	The activate counts more than cert token number

3.7 Checking Password

In certain cases system might have to validate user password without creating a login session. This could be used to eliminate unexpected termination to the login session when password validate is necessary. To check password without creating session, system could send the following command to SSM:

```
checkpwd|13|<user identity>|21|<password>|157|<device>|
```

Table 3.7.1 – SSM password checking attribute summary

Name	ID	Format	Meaning
OMS_USER_NO	13	Char	Account ID of the user. Case insensitive.
OMS_NAME	21	Char	Account password. Case sensitive.
OMS_DEVICENAME	157	Char	Name of device
OMS_MACHINE_ID	520	Char	Optional. Used for device with serial number.

An error message will be sent to the client after the password is validated. Please refer to the section 3.6 for the detail of error message.

3.8 Changing Password

This function is offered by SSM for certain situation. If it is opened, system could send the following message to have the login password changed. Session logon is not required for this function to work properly.

Password|13|<user identity>|21|<old password>|25|<new password>|22|<account type>|157|<device>|

Example:
password|13|1002|21|1234|25|4321|22|1|157|Internet|

Table 3.8.1 – SSM password changing attribute summary

Name	ID	Format	Meaning
OMS_USER_NO	13	Char	Account ID of the user. Case insensitive.
OMS_NAME	21	Char	Old password. Case sensitive.
OMS_FREE_TEXT	25	Char	New password. Case sensitive.
OMS_PRODTYPE	22	Char	Account type: 0=AE, 1=account (by default)
OMS_DEVICENAME	157	Char	Name of device

If the password changed successfully, the whole request message will be returned to client. In any case if the password validation failed, an error message will be return.

Please refer section 3.6 for detail of SSM error.

3.9 Changing PIN

If PIN number is required in OMS, it's allowed to be changed using below command

```
ChangePIN|13|<user identity>|22|<type>|117|<old PIN>|25|<new PIN>|157|<device>|
```

Example:

```
ChangePIN|13|1001|22|0|117|841562|25|489276|157|Internet|
```

Table 3.9.1 – SSM PIN changing attribute summary

Name	ID	Format	Meaning
OMS_USER_NO	13	Char	Account ID of the user. Case insensitive.
SSM_PIN	117	Char	Old PIN number. Case sensitive.
OMS_FREE_TEXT	25	Char	New PIN number. Case sensitive.
OMS_PRODTYPE	22	Char	Account type: 0=AE, 1=account (by default)
OMS_DEVICENAME	157	Char	Name of device

If the PIN number changed successfully, the whole request message will be returned to client. In any case if the PIN number validation failed, an error message will be return. Please refer section 3.6 for detail of SSM error.

3.10 Verifying PIN

If PIN number is required to finish the authentication, PIN number verify is needed, command as below

```
PIN|0|<session id>|13|<user identity>|117|<PIN>|157|<device>|
```

Example (PIN number here is 6 digit number “841562”, and requesting position at “1,2,6”):
PIN|0|0105225441012552221634604247707733337551022834821263072598144511|13|
CC0001@203.1.12.123|117|8,4,2|157|Internet|

Table 3.10.1 – PIN verifying attribute summary

Name	ID	Format	Meaning
OMS_SYMBOL	0	Integer(64)	Unique session ID
OMS_USER_NO	13	Char	User identity, formatted as <userid>@<host>
OMS_PIN	117	Char	The PIN number matched the PIN position, separated by comma
OMS_DEVICENAME	157	Char	Name of device

If verify successfully, the response will be like the one verifying session

```
session|0|<session id>|13|<user identity>|22|<user type>|116|<POS>|…
```

Table 3.10.2 – Verify PIN response attribute summary

Name	ID	Format	Meaning
OMS_SYMBOL	0	Integer(64)	Unique session ID
OMS_USER_NO	13	Char	User identity, formatted as <userid>@<host>
OMS_PRODTYPE	22	Char	Account type: 0=AE, 1=Trade
OMS_MATURITY	29	DateTime	Password expiry date time, in format “yyyy/MM/dd HH:mm:ss”
OMS_PIN_POS	116	Char	Random PIN number positions (optional),

Name	ID	Format	Meaning
			separated by comma
OMS_PIN_EXPIREDATE	118	DateTime	PIN number expiry date, in format “yyyy/MM/dd HH:mm:ss”
OMS_TRADE_DATE	134	DateTime	Server side date time, in format “yyyy/MM/dd HH:mm:ss”
OMS_FIRSTROW	200-299	Char	User Entitlement (optional)
OMS_LAST_EXETIME	1115	DateTime	Last logon time, in format “yyyy-MM-dd HH:mm:ss”

But if verify failed, error message will be received

error|0|<session id>|13|<user identity>|25|<error message>|39|<error code>|

Example:

```
error|0|0105225441012552221634604247707733337551022834821263072598144511|13|
CC0001@203.1.12.123|25|PIN number verify failed|39|70|
```

Please refer section 3.6 for detail of SSM error.

4 iTrader Server Operation

4.1 Overview

Instructions to iTS are divided into four different classes, they are session verification, account information querying, order server instructions and distributed data subscription. The first class is discussed in the previous chapter. All of the other classes of instructions will only be accepted when the login session is verified.

All instruction, apart from session verification is apply to trade account only. If a user logon the system using an AE account, all instruction should include the trade account ID, which is normally attribute “10”, before the instruction can be processed property. Note that the trade account will be ignored if the login session is corresponding to a trade account. Error will be generated if the specified trade account is not belonging to the AE account of the verified login session.

The general instruction message format of iTS is shown below:

```
<System directive>|<instruction>|<attr1>|<value1>|<attr2>|value2|…
```

Example:

```
ORD|calculate|0|00001|4|1000|3|105|13|CC0001|400|I|13|8820|
```

System directive is the name server component going to receive the instruction. It can be “ITS” or “ORD”. Always use capital letters for these directives. iTS will filter and route messages to corresponding server component and catch the feedback for the client application.

ITS introduces multi-thread mechanism for handling client requests. Every Client request will push into thread pool for farther process. And result message will be return once process completed. Because ITS is using multi-thread, the feedback message not guarantee response in sequence. For example, client application sends command to get account position and then get trade.

It may be return trade faster than position. Such as client sends below commands

```
<1> ITS|position|
<2> ITS|querysuborder|
```

And ITS may return querysuborder result faster than position result

```
querysuborder|1/2|6|110427000001|7|BMFTS_E141125:1|0|00001|4|1000|3|59|13|S001|10|CC0001|20|SEHK|501|14:11:25|47|0|15|1|101|0|43|0|5|3|78||33|14:11:25|

querysuborder|2/2|6|110427000002|7|BMFTS_E141215:3|0|00002|4|1000|3|37.1|13|S001|10|CC0001|20|SEHK|501|14:12:15|47|0|15|1|101|0|43|0|5|3|78||33|14:12:15|

position|1/3|10|CC0001|0|00001|38|-59000|4|1000|115||
position|2/3|10|CC0001|0|00002|38|-37100|4|1000|115||
position|3/3|10|CC0001|0|00006|38|-31000|4|1000|115||
```

Developer need check the header to identity the return message is for which request.

Client application may need to show the trading account’s position and balance, for this purpose, account’s position is built on BOD position and trades, detail please refer to section Account Position, while for account balance, it’s maintained by OMS and client application should subscribe for it, using the commands defined in section General Account Information.

4.2 Session Maintenance

A login session will expire if there is no verification operation on certain period of time (default is 15 minutes). It is an important duty for a client application to keep the login session active. The timer will be renewed if a verification message of that particular session is received before it is expired.

The following message instruct ITS to verify SSM, for keeping the session active.

```
ITS|verifysession|0|<session id>|
```

ITS will help to send the verify message to SSM, and there is not reply message to client for this command.

4.3 General Account Information

The following message instruct ITS to return the general account information to the client application.

```
ITS|queryaccount|[10|<trade account id>|]
```

Example: ITS|queryaccount|10|CC0001|

The return message will be formatted as follow:

```
queryaccount|155|<trade limit>|156|<account balance>|…
```

Example: queryaccount|155|199900.00|156|1300|185|199900.00|186|1300|

Table 4.3.1 – Query account information attribute summary

Image Tag for Stock Account

Name	ID	Format	Meaning
OMS_ACCOUNT	10	Char	Account ID
OMS_TRADE_LIMIT	155	Float	Account trading limit
OMS_CASH_BALANCE	156	Float	Account balance
OMS_BOD_TRADE_LIMIT	185	Float	Account begin of day trading limit
OMS_BOD_CASH_BALANCE	186	Float	Account begin of day cash balance

Image Tag for Future Account

Name	ID	Format	Meaning
OMS_ACCOUNT	10	Char	Account ID
OMS_TRADE_LIMIT	155	Float	Account trading limit
OMS_CASH_BALANCE	156	Float	Account balance
OMS_BOD_TRADE_LIMIT	185	Float	Account begin of day trading limit
OMS_BOD_CASH_BALANCE	186	Float	Account begin of day cash balance
OMS_REQUIREDMARGIN	604	Float	Initial Margin

Name	ID	Format	Meaning
OMS_MNTMARGIN	606	Float	Maintenance Margin
OMS_PL	607	Float	P&L
OMS_MARGINCALL	996	Float	Surplus/Margin Call

Client application can make use of this account information to build up the account balance, by taking omsTag#155 as current trading limit, while omsTag#156 as current cash balance.

4.4 Multi-Currency Account Information

The following message instruct ITS to subscribe the multi-currency account information.

ITS|currencyaccount|[10|<trade account id>|]

Example: ITS|currencyaccount |10|CC0001|

The return message will be formatted as follow:

currencyaccount|23|<currency>|155|<trade limit>|156|<account balance>|…

Example:

currencyaccount |23|HKD|155|199900.00|156|1300|185|199900.00|186|1300|

currencyaccount |23|USD|155|299900.00|156|2300|185|299900.00|186|2300|

Table 4.4.1 – multi-Currency account information attribute summary

Image Tag for Stock Account

Name	ID	Format	Meaning
OMS_ACCOUNT	10	Char	Account ID
OMS_CURRENCY	23	Char	Currency code, e.g. ‘HKD’
OMS_TRADE_LIMIT	155	Float	Account trading limit
OMS_CASH_BALANCE	156	Float	Account balance
OMS_BOD_TRADE_LIMIT	185	Float	Account begin of day trading limit

OMS_BOD_CASH_BALANCE	186	Float	Account begin of day cash balance

Image Tag for Future Account

Name	ID	Format	Meaning
OMS_ACCOUNT	10	Char	Account ID
OMS_CURRENCY	23	Char	Currency code, e.g. 'HKD'
OMS_TRADE_LIMIT	155	Float	Account trading limit
OMS_CASH_BALANCE	156	Float	Account balance
OMS_BOD_TRADE_LIMIT	185	Float	Account begin of day trading limit
OMS_BOD_CASH_BALANCE	186	Float	Account begin of day cash balance
OMS_REQUIREDMARGIN	604	Float	Initial Margin
OMS_MNTMARGIN	606	Float	Maintenance Margin
OMS_PL	607	Float	P&L
OMS_MARGINCALL	996	Float	Surplus/Margin Call

Client application can make use of this account information to build up the multi-currency account balance, by taking omsTag#23 as currency code, omsTag#155 as currency trading limit, while omsTag#156 as currency cash balance.

4.5 Account Position

The following message instruct iTS to return stock position of the logon trade account.

```
ITS|position|[10|<trade account id>|]
```

Example: ITS|position|10|CC0001|

The response will be in multiple messages representing position of different stock.

```
position|<index>|<attr1>|<value1>|<attr2>|<value2>|…
```

Example: position|1/1|10|CC0001|0|00001|38|75000.975|4|7000|

Table 4.5.1 – Position query attribute summary

Name	ID	Format	Meaning
OMS_SYMBOL	0	Char	Stock symbol
OMS_QUANTITY	4	Float	Long Quantity
OMS_ACCOUNT	10	Char	Trade account ID
OMS_TURNOVER	38	Float	Amount (not to be used)
OMS_AVG_PRICE	115	Float	Average Price

As the reply from querying position is in multiple messages, iTS include an index field format in "<index>/<total>" indicating the sequence and total number of record return. If no position record found, an error will be generated.

```
error|9|POSITION|25|Record not found|39|130|
```

Besides the position command, also "boddata" command is designed for BOD positions' query for particular account.

```
ITS|boddata|[10|<trade account id>|]
```

Example: ITS|boddata|10|CC0001|

The "boddata" command works like "position", but it'll return one more field "short quantity" in field 470

Example: boddata|1/7|10|CC0001|0|00005|38|0|4|4000|470|0|115|116|

Table 4.5.2 – BOD Position query attribute summary

Name	ID	Format	Meaning
OMS_SYMBOL	0	Char	Stock symbol
OMS_QUANTITY	4	Float	Long Quantity
OMS_ACCOUNT	10	Char	Trade account ID
OMS_TURNOVER	38	Float	Amount (not to be used)
OMS_AVG_PRICE	115	Float	BOD Average Price

OMS_SHORT_QUANTITY	470	Integer	Short Quantity

4.6 Order History

Client application can query account order history by sending message to iTS in 2 format:

Format 1: get history order by days

```
ITS|history|4|<number of days>|[10|<trade account id>|]
```

Example: ITS|history|4|30|10|CC0001|

Client application should specify the attribute “4” with the number of days to query.

Format 2: get history order by date range

```
ITS|history|10|<Account ID>|1|<Start Date yyyy/MM/dd>|2|<End Date yyyy/MM/dd >|
```

Example: ITS|history|10|CC0001|1|2013/03/01|2|2013/04/11|

Client application should specify the attribute “1” with start date, “2” with end date to query.

Just like querying account position, the response will be in multiple messages. Each message contains the information about an order in the history record.

```
history|<index>|<attr1>|<value1>|<attr2>|<value2>|
```

Example:

```
history|1/1|10|CC0001|0|00001|4|1000|11|0|6|10000159|3|9.85|5|Inac|34|0|13|8802
|42|0|33|14:32:14|134|2001/04/27|400|I|
```

Table 4.6.1 – Order history query attribute summary

Name	ID	Format	Meaning
OMS_SYMBOL	0	Char	Stock symbol
OMS_PRICE	3	Float	Limited price
OMS_QUANTITY	4	Float	Quantity
OMS_STATUS	5	Char	Order status "Pend" - Pending "Part" - Partial completed "Comp"- Completed

			"Canc" - Cancelled "Inac" - Inactive "Chec" - Checked "Conf" - Confirmed "Reje" - Rejected
OMS_ORDER_NO	6	Integer	Order number (Identity of a particular order)
OMS_ACCOUNT_NO	10	Char	Trade account ID
OMS_ORDER_TYPE	11	Integer	Order type: 0=Buy, 1=Sell
OMS_TIME	33	Time	Ordering time
OMS_FILLQTY	34	Integer	Fill quantity
OMS_PRICE_2	42	Float	Second Price (Invalid field- should not be used)
OMS_TRADE_DAY	134	Date	Ordering Date
OMS_OPERATORFLAG	400	Char	Operator Flag

As the reply from querying order history is in multiple messages, iTS include an index field format in “<index>/<total>” indicating the sequence and total number of record return. If no position record found, an error will be generated.

4.7 Custom Information Query

Apart from the query defined above, ITS also offer a flexible interface for custom information query. This enable ITS to be configured for different environment to provide proper information to connecting client accordingly. The general message for requesting custom query is defined as following:

```
Customquery|457|<queryid>|[[<attr1>|<value1>|]<attr2>|<value2> …]
```

Table 4.7.1 – Custom information query attribute summary

Name	ID	Format	Meaning
OMS_SYSTEMREF	457	Char	An ID identifying the requesting query

The request message contains a required field OMS_SYSTEMREF which if defined to be an identity of the query requesting. It might followed by a number of attribute-value pairs according to the query defined. The required attribute-value pair for specific query will be provided in another document tailor made for the application.

Like other ITS query request, the query result will be return by one or more message, once per record found, in the following format:

```
customquery|<index>|<attr1>|<value1>|<attr2>|<value2>|
```

For the same reason, the meaning of all returning attribute-value pairs will be defined in another document tailor made for the application of the query.

4.8 Suborder Query

Client application can query information of trades completed of the current session by sending message to iTS in the following format:

```
ITS|querysuborder|[10|<trade account id>|]
```

Example: ITS|querysuborder|10|CC0001|

Similar to querying account position, the response will be in multiple messages. Each message contains the information about a record of trade information.

```
querysuborder|<index>|<attr1>|<value1>|<attr2>|<value2>|
```

Table 4.8.1 – Suborder query attribute summary

Name	ID	Format	Meaning
OMS_SYMBOL	0	Char	Stock symbol
OMS_L_PRICE	3	Float	Exchange order price
OMS_QUANTITY	4	Float	Exchange order quantity
OMS_STATUS	5	Integer	Exchange order status
OMS_ORDER_NO	6	Char	OMS order ID
OMS_EXCH_NO	7	Char	Exchange order ID
OMS_ACCOUNT	10	Char	Trade account ID
OMS_USER_NO	13	Char	User ID. Usually the ID of the AE
OMS_ORIGINAL	15	Char	Linked exchange order ID
OMS_EXCHANGE	20	Char	Exchange ID
OMS_TIME	33	Time	Last updated time
OMS_TRADETYPE	43	Char	Trading type (Obsolete)
OMS_HEDGE	47	Char	Hedge ratio (Obsolete)
OMS_EXCHANGEDEST	78	Char	Destination exchange ID

Name	ID	Format	Meaning
OMS_SHORTSELL	101	Integer	Short sell flag
OMS_CREATETIME	501	Time	Creation time

As the reply from querying trade is in multiple messages, iTS include an index field format in “<index>/<total>” indicating the sequence and total number of record return. If no trade record found, an error of “record not found” will be signaled, format as below

```
error|9|QUERYSUBORDER|25|Record not found|39|130|
```

4.9 Trade Query

Client application can query information of trades completed of the current session by sending message to iTS in the following format:

```
ITS|querytrade|[10|<trade account id>|]
```

Example: ITS|querytrade|10|CC0001|

Similar to querying account position, the response will be in multiple messages. Each message contains the information about a record of trade information.

```
querytrade|<index>|<attr1>|<value1>|<attr2>|<value2>|
```

Table 4.9.1 – Trade query attribute summary

Name	ID	Format	Meaning
OMS_SYMBOL	0	Char	Stock symbol
OMS_L_PRICE	3	Float	Price completed
OMS_QUANTITY	4	Float	Trade quantity
OMS_STATUS	5	Integer	Trade status
OMS_EXCH_NO	7	Char	Exchange number
OMS_TRAN_NO	8	Char	Transaction ID
OMS_ACCOUNT	10	Char	Trade account ID

OMS_ORDER_TYPE	11	Integer	Order type: 0=Buy, 1=Sell
OMS_USER_NO	13	Char	User ID. Usually the ID of the AE
OMS_TIME	33	Time	Order completion time

As the reply from querying trade is in multiple messages, iTS include an index field format in "<index>/<total>" indicating the sequence and total number of record return. If no trade record found, an error of "record not found" will be signaled, format as below:

```
error|9|QUERYTRADE|25|Record not found|39|130|
```

4.10 Money Voucher Query

Client application can query account money voucher by sending message to iTS in the following format:

```
ITS|querymoneyvoucher|[10|<trade account id>|]
```

Example: ITS|querymoneyvoucher|10|CC0001|

Similar to querying account position, the response will be in multiple messages. Each message contains the information about a record of voucher information.

```
querymoneyvoucher|<index>|<attr1>|<value1>|<attr2>|<value2>|
```

Table 4.10.1 – Query money voucher attribute summary

Name	ID	Format	Meaning
OMS_SYMBOL	0	Char	Currency symbol, e.g. 'HKD'
OMS_L_PRICE	3	Float	Voucher amount
OMS_QUANTITY	4	Integer	Voucher quantity, always 1 for money voucher
OMS_STATUS	5	Integer	Voucher Status
OMS_ORDER_NO	6	Char	Order number (Identity of a particular order)
OMS_EXCH_NO	7	Char	Exchange number
OMS_TRAN_NO	8	Char	Transaction ID

OMS_ACCOUNT	10	Char	Trade account ID
OMS_ORDER_TYPE	11	Integer	Order type: 0=Withdrawal, 1=Deposit
OMS_USER_NO	13	Char	User ID. The ID of the settlement staff
OMS_TIME	33	Time	Order completion time

As the reply from querying money voucher is in multiple messages, iTS include an index field format in "<index>/<total>" indicating the sequence and total number of record return. If no trade record found, an error of "record not found" will be signaled

```
error|9|QUERYMONEYVOUCHER|25|Record not found|39|130|
```

4.11 ITS Error Message

Error message will be sent to client application when the instruction is invalid or requesting operation executed failed. All error messages should be formatted as follow:

```
Error|25|<error message>|39|<error code>|
```

Example: error|25|Command unknown|39|110|

Table 4.11.1 – Defined ITS error message summary

Code	Message	Possible reason
110	Command unknown	Specified system directive invalid
111	ITS command unknown	Specified iTS instruction invalid
112	Verification required	User attempt to execute instruction before session verification
113	Cipher key not set	Encryption key is not initialized
114	Verification duplicated	Double account verify in iTS authentication
115	Order service not available	Order service not available
116	Address access denied	Address ID provided is invalid
120	Account ID invalid	Trade account is not specified or not valid for the

Code	Message	Possible reason
		instruction while user logon the system as AE user. Trade account specified is not belong to the AE user.
121	Invalid number of days	Number of days not specified correctly while querying order history
122	Cipher Key Invalid	Encryption key initialization failed
123	Trading password failed	Trading password verify failed for ORD commands
130	Record not found	Account information record not found
131	Settlement failed	Settlement notification failed
132	Market close	Market session expired for trading command.

5 Order Interface

5.1 Overview

All order related instructions will be sent to order server ORD through iTS. Client application should send all these instructions to iTS with the system directive “ORD”.

```
ORD|<instruction to order server>|[510|<trading password>]
```

Depending on account setting, all messages routing to ORD might require a trading password for verification. Fail to provide a required trading password would result in operation denied.

Any feedback from ORD and daily order book of trade accounts will be distributed by DDS. However, iTS will subscribe those information for any connected client after user’s login session is verified successfully. So client application should expect to receive a full image of order book of corresponding trade account and updates on particular order entry occasionally.

Please note that all message summary list only some meaningful attributes. Client application should ignore other attributes in the message it received.

5.2 Order Book

A full image of order book will be sent to client application upon successful verification. Each order entry will be formatted as a single message in the following format:

image|ORD|<attr1>|<value1>|<attr2>|<value2>|

Example:

image|ORD|6|10000017|0|00036|10|CC0001|11|0|13|1001|4|2000|3|10.000|5|3|33|
10:40:50|501|10:38:59|34|2000|35|0|42|10.000|20|SEHK|502|TRADE|70|-
20000.000|71|-20000.000|401|2000|23|HKD|

Table 5.2.1 – Order book image attribute summary

Name	ID	Format	Meaning
OMS_SYMBOL	0	Char	Stock symbol
OMS_PRICE	3	Float	Limited price
OMS_QUANTITY	4	Float	Quantity
OMS_STATUS	5	Integer	Order status: 1. Pending 2. Partial completed 3. Completed 4. Cancelled 5. Inactive 6. Checked 7. Confirmed -1. Rejected
OMS_ORDER_NO	6	Char	Order number (Identity of a particular order)
OMS_ACCOUNT	10	Char	Trade account ID
OMS_ORDER_TYPE	11	Integer	Order type: 0=Buy, 1=Sell
OMS_USER_NO	13	Char	Corresponding AE account ID
OMS_EXCHANGE	20	Char	Name of exchange
OMS_FREE_TEXT	25	Char	Free text comments on this order entry

Name	ID	Format	Meaning
OMS_TIME	33	Time	Ordering time
OMS_FILLQTY	34	Integer	Filled quantity
OMS_ERRORCODE	39	Integer	Error code
OMS_PRICE_2	42	Float	Average price
OMS_CREDIT	70	Float	Account credit
OMS_QUEUE_QTY	122	Integer	Queue quantity
OMS_OPERATORFLAG	400	Char(1)	Operator Flag indicating where the order entry from.
OMS_WORKING	401	Integer	Working quantity
OMS_MINORCODE	464	Integer	Minor code
OMS_CREATETIME	501	Time	Order creation time
OMS_RESOURCE	502	Char	Resource ID

Notes:

1. Order Status “Pending” implies either the order is stored in the system or already sent out to the exchange for queuing. But if order queue quantity > 0, the order is queuing in the exchange. Otherwise it is stored in the system.

Client application should use the order number (attribute “6”) as a unique identifier of each order entry. Whenever there are new order entries or information changes of an existing order, a completed image of that particular order will be sent to client application in the same format as above. So, client application can keep the order book updated.

5.3 Placing Order

Client application can place an order for a trade account. If the user login a session of a trade account, iTS will ignore the account ID (attribute "10") and override it with corresponding trade account from the login session. On the other hand, if the user login a session of an AE account, it is necessary to specify a trade account the order instruction entitled to. Here is the message format for placing order:

```
ORD|add|<attr1>|<value1>|<attr2>|<value2>|…
```

Example: ORD|add|0|00035|10|CC0017|11|0|13|1001|4|2000|3|10.400|400|I|

Table 5.3.1 – Order placing instruction attribute summary

Name	ID	Format	Req'd	Meaning
OMS_SYMBOL	0	Char(15)	Y	Stock symbol
OMS_PRICE	3	Float	Y	Limited price
OMS_QUANTITY	4	Float	Y	Quantity
OMS_ACCOUNT	10	Char	Y	Trade account ID
OMS_ORDER_TYPE	11	Integer	Y	Order type: 0=Buy, 1=Sell
OMS_USER_NO	13	Char	Y	The user ID placing this order
OMS_INSTRUCT	40	Char	N	Order instruction* *see table below*
OMS_PRICE1	41	Float	N	Price for special instruction (optional)
OMS_PRICE2	42	Float	N	Average Price
OMS_USER_REF	74	Char(50)	N	Free text used for client's reference (optional)
OMS_SHORTSELL	101	Char	N	Short Selling Flag: 1=Yes, 0=No (optional)
OMS_SESSION_KEY	191	Char	Y	The session key of the logon session, used to identify the session of the order
OMS_OPERATORFLAG	400	Char	Y	Channel ID, if there's no custom, should be "I"

Name	ID	Format	Req'd	Meaning
OMS_BASKET_NO	455	Char	N	Basket or bulk parent order ID (optional)
OMS_COMP_SYSREF	487	Char	N	System reference (optional), comma separated string list in format "Key0=Value0,Key1=Value1,…"
OMS_INSTRUCT_1	489	Char	N	Order instruction1 (optional)
OMS_INSTRUCT_2	490	Char	N	Order instruction2 (optional)
OMS_PASSWORD	510	Char	Y	Trading password

Notes:

1. For auction orders which are unfilled, the field 40 will be changed from 512 to 544.
2. For stop limit orders, field 3 and field 41 are required. For stop market orders, field 41 is required.

If the order placed successfully, the image of the new order information with the unique order number will be sent to client application in the form of updating order book.

Table 5.3.2 - OMS_INSTRUCT (FIELD ID 40) TABLE

Order Type	Value to be used
Limit Order	0 (by default)
Odd Lot Order	3
Fill And Kill	128
Fill Or Kill	256
Auction Order	512
Stop Limit Order	2048

More samples for reference

1. Place new order

```
image|ORD_10000019|6|10000019|0|00006|10|X1|11|0|13|2001|4|4000|3|31.000|5|1|33|19:15:5
4|501|19:15:54|34|0|35|0|20|SEHK|43|0|44|0|187|add|191|SSMS44baOJaJBmI5OaZq|1109|2|70|-
124000.000|488|I|502|TRADE|71|-124000.000|95|-
124000.000|401|4000|400|D|23|HKD|461|125|464|0|1116|2|
```

Exchange acknowledged:

```
image|ORD_10000019|6|10000019|0|00006|10|X1|11|0|13|2001|4|4000|3|31.000|5|1|33|19:15:5
4|501|19:15:54|34|0|35|0|20|SEHK|43|0|44|0|187|add|191|SSMS44baOJaJBmI5OaZq|1109|3|70|-
124000.000|488|I|502|TRADE|71|-124000.000|95|-
124000.000|401|4000|400|D|23|HKD|461|125|464|0|1116|2|
```

2. Exchange order received due to 1st execution

```
image|ORD_10000019|6|10000019|0|00006|10|X1|11|0|13|2001|4|4000|3|31.000|5|2|33|19:15:5
4|501|19:15:54|34|2000|35|0|20|SEHK|43|0|44|0|187|add|191|SSMS44baOJaJBmI5OaZq|1109|4|7
0|-124000.000|488|I|502|TRADE|71|-124000.000|95|-
124000.000|401|4000|400|D|23|HKD|461|125|464|0|1116|2|
```

Trade received

```
image|ORD_10000019|6|10000019|0|00006|10|X1|11|0|13|2001|4|4000|3|31.000|5|2|33|19:15:5
4|501|19:15:54|34|2000|35|0|42|31.000000|20|SEHK|43|0|44|0|187|add|191|SSMS44baOJaJBmI5
OaZq|1109|4|70|-124000.000|488|I|502|TRADE|71|-124000.000|95|-
124000.000|401|4000|400|D|23|HKD|461|125|464|0|1110|31.000|1111|2000.000|1112|3|1113|11
10539754580092|1115|19:15:54|1116|2|
```

3. Exchange order received due to 2nd execution

```
image|ORD_10000019|6|10000019|0|00006|10|X1|11|0|13|2001|4|4000|3|31.000|5|3|33|19:15:5
4|501|19:15:54|34|4000|35|0|42|31.000000|20|SEHK|43|0|44|0|187|add|191|SSMS44baOJaJBmI5
OaZq|1109|5|70|-124000.000|488|I|502|TRADE|71|-124000.000|95|-
124000.000|401|4000|400|D|23|HKD|461|125|464|0|1116|2|
```

Trade received

```
image|ORD_10000019|6|10000019|0|00006|10|X1|11|0|13|2001|4|4000|3|31.000|5|3|33|19:15:5
4|501|19:15:54|34|4000|35|0|42|31.000000|20|SEHK|43|0|44|0|187|add|191|SSMS44baOJaJBmI5
OaZq|1109|5|70|-124000.000|488|I|502|TRADE|71|-124000.000|95|-
124000.000|401|4000|400|D|23|HKD|461|125|464|0|1110|31.000|1111|2000.000|1112|3|1113|11
10539754588536|1115|19:15:54|1116|2|
```

5.4 Calculate Fee & Charges

Client application may need to know the consideration amount if placing an order, and the “calculate” command is designed for this purpose, allowing to pre-calculate the charges if placing an order with given price and quantity, and the command is defined as below

```
ORD|calculate|0|<symbol>|11|<B/S>|3|<price>|4|<quantity>|10|<userid>|400|I|23|<currency>|
```

Example:
ORD|calculate|0|00008|11|0|3|4.975|4|1000|10|CC0001|13|103|400|I|23|HKD|

Table 5.4.1 – Calculate fees/commission attribute summary

Name	ID	Format	Req’d	Meaning
OMS_SYMBOL	0	Char(15)	Y	Stock symbol
OMS_PRICE	3	Float	Y	Limited price
OMS_QUANTITY	4	Float	Y	Quantity
OMS_ACCOUNT	10	Char	Y	Trade account ID
OMS_ORDER_TYPE	11	Integer	Y	Order type: 0=Buy, 1=Sell
OMS_USER_NO	13	Char	Y	Order user ID
OMS_CURRENCY	23	Char	N	Symbol currency
OMS_OPERATORFLAG	400	Char(1)	N	Operator Flag indicating where the order entry from.

The following message format shows commission and fees details:

```
image|CALCD_<userid>|0|<symbol>|10|<accountid>|11|<B/S>|13|<userid>|4|<quantity>|3|<price>|400|I|23|<currency>|406|<Order Amount w/ commissions & fees>|402|<Commission >|403|<Stamp Duty>|404|<Levy>|100|<CCASS Fees>|405|<Total fees>|
```

Example:
image|CALCD_2002|0|00008|10|CC0001|11|0|13|2002|4|1000|3|4.975|23|HKD|406|4975.000|402|100.000|403|5.000|404|0.600|100|5.000|405|10.600|

Table 5.4.2 – Calculate fees/commission return message format

Name	ID	Format	Meaning
OMS_CCASS	100	Float	CCASS fees
OMS_COMMISSION	402	Float	Commission
OMS_STAMPDUTY	403	Float	Stamp Duty
OMS_LEVY	404	Float	Levy
OMS_TOTAL	405	Float	Total Fees = CCASS Fees + Stamp Duty + Levy
OMS_TRDVALUE	406	Float	Order Amount excluding fees and commission

Please note that the calculation won’t hold fund for the trading account, but just shows the charges.

5.5 Changing Order

Change order instruction can be sent when the market is open and the order status is pending or partial filled.

Client application is able to change parameters of uncompleted order by following message:

```
ORD|change|<attr1>|<value1>|<attr2>|<value2>|…
```

Example: ORD|change|6|1000000001|13|1001|4|2000|3|10.400|

Table 5.5.1 – Order changing instruction attribute summary

Name	ID	Format	Req'd	Meaning
OMS_PRICE	3	Float	N	Limited price (optional)
OMS_QUANTITY	4	Float	N	Quantity (optional)
OMS_ORDER_NO	6	Char	Y	Order number identifying order to change
OMS_USER_NO	13	Char	Y	The user ID who change the order
OMS_USER_REF	74	Char(50)	N	Free text used for client's reference (optional)
OMS_SESSION_KEY	191	Char	Y	The session key of the logon session, used to identify the session of the order
OMS_OPERATORFLAG	400	Char	N	Channel ID, should be the same one with placing order
OMS_PASSWORD	510	Char	Y	Trading password

Just like placing a new order, if the changes applied successfully, the updated image will be sent to client in the form of updating order book.

5.6 Cancelling Order

Client application is able to cancel uncompleted order (with order status “Pending”, “Partial” or “Inactive”) by following message:

```
ORD|cancel|<attr1>|<value1>|<attr2>|<value2>|…
```

Example: ORD|cancel|6|1000000001|13|1001|

Table 5.6.1 – Order canceling instruction attribute summary

Name	ID	Format	Req’d	Meaning
OMS_ORDER_NO	6	Char	Y	Order number identifying order to cancel
OMS_USER_NO	13	Char	Y	The user ID who cancel the order
OMS_USER_REF	74	Char(50)	N	Free text used for client’s reference (optional)
OMS_SESSION_KEY	191	Char	Y	The session key of the logon session, used to identify the session of the order
OMS_OPERATORFLAG	400	Char	N	Channel ID, should be the same one with placing order
OMS_PASSWORD	510	Char	Y	Trading password

Just like placing changing an order, if the order canceled successfully, the updated image, marked order status with “cancel”, will be sent to client in the form of updating order book.

5.7 Error Handling

Apart from order book, iTS will also subscribe the error event notification from DDS. Error event occur when order server unable to handle the requested operation for some reasons, such as that the exchange is rejected the order. Error event notification will be sent to client in the following format:

```
image|ERRORD_<userid>|<attr1>|<value1>|<attr2>|<value2>|…
```

Example: image|ERRORD_103|5|-1|25|ORD-7 Price is not at spread level|33|10:00:00|39|-7|464|-70000|

Table 5.7.1 – Error Order attribute summary

Name	ID	Format	Meaning
OMS_STATUS	5	Integer	Order latest status
OMS_FREE_TEXT	25	Char	Error message from exchange
OMS_TIME	33	Time	Status time from exchange
OMS_ERRORCODE	39	Integer	Error code
OMS_SESSION_KEY	191	Char	The session key of the logon session, used to identify the session of the order
OMS_MINORCODE	464	Integer	Minor code

Please refer to Appendix 4 for the possible error code and minor code defined in OMS.

Please note that, the error order image is based on user, which means, if several errors happened on the same user, only the last error order image will be available in DDS, or in other words, client application can only subscribe the last error order image.

The “Session Key” here can be used to identify if an order error was triggered in current logon session, so that client application can judge if error message popup is needed, in order to avoid popup an “old” error.

Appendix 1 – List of Attributes

Name	ID	Format	Meaning
OMS_SYMBOL	0	Char	Stock symbol
OMS_PRICE	3	Float	Limited price
OMS_QUANTITY	4	Float	Quantity
OMS_STATUS	5	Integer	Order latest status
OMS_ORDER_NO	6	Integer	Order number (Identity of a particular order)
OMS_EXCH_NO	7	Integer	Exchange order ID (Exchange level)
OMS_TRAN_NO	8	Char	Transaction ID
OMS_ACCOUNT	10	Char	Trade account ID
OMS_ORDER_TYPE	11	Integer	Order Type 0=Buy, 1=Sell:
OMS_USER_NO	13	Char	Corresponding AE account ID
OMS_EXCHANGE	20	Char	Name of exchange
OMS_NAME	21	Char	Old password. Case sensitive.
OMS_PRODTYPE	22	Char	Account type: 0=AE, 1=account (by default)
OMS_FREE_TEXT	25	Char	New password. Case sensitive.
OMS_TIME	33	Time	Ordering time
OMS_FILLQTY	34	Integer	Fill quantity
OMS_TURNOVER	38	Float	Amount (not to be used)
OMS_ERRORCODE	39	Integer	Error code
OMS_INSTRUCT	40	Char	Order instruction* *see table below*
OMS_PRICE1	41	Float	Price for special instruction (optional)
OMS_PRICE2	42	Float	Average Price (should not be used in HISTORY)
OMS_CREDIT	70	Float	Account credit
OMS_USER_REF	74	Char(50)	Free text used for client’s reference (optional)

OMS_EXCHANGEDEST	78	Char	Exchange Dest.
OMS_CCASS	100	Float	CCASS fees
OMS_SHORTSELL	101	Char	Short Selling Flag: 1=Yes, 0=No (optional)
OMS_TRADE_DAY	134	Date	Ordering Date
OMS_TRADE_LIMIT	155	Float	Account trading limit
OMS_CASH_BALANCE	156	Float	Account balance
OMS_DEVICENAME	157	Char	Name of device
OMS_SETTLEMENT_TYPE	158	Char(8)	Custom message type string
OMS_MSG_REF_ID	161	Char	Reference ID of the received messagee
OMS_MSG_RECIPIENT	162	Char(200)	The recipient ID the message sending to
OMS_MSG_SUBJECT	163	Char(200)	The Subject of the message
OMS_BOD_TRADE_LIMIT	185	Float	Account begin of day trading limit
OMS_BOD_CASH_BALANCE	186	Float	Account begin of day cash balance
OMS_OPERATORFLAG	400	Char(1)	Operator Flag indicating where the order entry from.
OMS_WORKING	401	Integer	Working
OMS_COMMISSION	402	Float	Commission
OMS_STAMPDUTY	403	Float	Stamp Duty
OMS_LEVY	404	Float	Levy
OMS_TOTAL	405	Float	Total Fees = CCass Fees + Stamp Duty + Levy
OMS_TRDVALUE	406	Float	Order Amount excluding fees and commission
OMS_BOD_TRDFUND	471	Float	BOD tradable fund
OMS_TRDFUND	472	Float	Tradable fund
OMS_BOD_WITHDRAWFUND	473	Float	BOD withdrawal fund
OMS_WITHDRAWFUND	474	Float	Withdrawal fund
OMS_BOD_TRDSTOCKBAL	475	Float	BOD tradable stock balance
OMS_TRDSTOCKBAL	476	Float	Tradable stock balance

OMS_BOD_WITHDRAWSTOCKBAL	477	Float	BOD withdrawal stock balance
OMS_WITHDRAWSTOCKBAL	478	Float	Withdrawal stock balance
OMS_TODAYBUYQTY	479	Integer	Today BUY Quantity
OMS_TODAYSELLQTY	480	Integer	Today SELL Quantity
OMS_PENDBUYQTY	481	Integer	Pending BUY Quantity
OMS_PENDSELLQTY	482	Integer	Pending SELL Quantity
OMS_DEPOSITQTY	483	Integer	Stock Deposit Quantity
OMS_WITHDRAWQTY	484	Integer	Stock Withdraw Quantity
OMS_INSTRUCT_1	489	Integer	Order instruction
OMS_INSTRUCT_2	490	Integer	Order instruction
OMS_CREATETIME	501	Time	Order creation time
OMS_RESOURCE	502	Char	Resource ID
OMS_MACHINE_ID	520	Char	Optional. Used for device with serial number.
OMS_LAST_PRICE	1110	Float	Trade Price
OMS_LAST_QUANTITY	1111	Float	Trade Qty
OMS_LAST_STATUS	1112	Char	Last Status
OMS_LAST_TRDNUM	1113	Char	Trade ref no
OMS_LAST_EXETIME	1115	Time	Exec time

Appendix 2A – Order Management during Non-Continuous Session

Before 9:00 a.m.	New orders can be entered, modified and cancelled after a preset time (for trading days only). Upon market pre-open at 9:00 am these orders will be sent to the Exchange automatically. Please refer to the table for pre-open session order behavior.
12:00p.m.-12:30 p.m.	New orders can be entered, modified and cancelled. Upon market open at 1:00 pm these orders will be sent to the Exchange automatically.
12:30 p.m.-1:00p.m.	New orders can be entered, modified and cancelled. Orders already sent to the Exchange during morning session can be cancelled.
4:00 p.m.	No more orders will be accepted by the System upon day end market close. Setting 1: these orders will be pending in the system. Setting 2: these orders will be rejected.

Appendix 2B – Order Behavior during Pre-open Session

4 Periods	Order Types	Action
1. Order Input / Pre-opening (9:00 – 9:15 a.m.)	● Limit orders ● Auction orders	Both Order types can be placed, changed and cancelled.
2. Pre-order Matching / Pre-open allocation (9:15 – 9:20 a.m.)	● Auction orders	- Can be placed only; - Change or cancellation shall be rejected.
	● Limit orders	- Will be pending in the system to be sent out automatically upon market open, can be changed or cancelled. * For orders queuing in the Exchange, no change or cancellation is permitted.
3. Order Matching /Open allocation (9:20 – 9:28 a.m.)	● Auction orders	Order placement, change and cancellation rejected.
	● Limit orders	- same behavior as in Pre-order Matching
	● Matching behavior	- Auction orders will be matched first. Unfilled quantity will be cancelled by the Exchange. - Unfilled limit orders will be carried forward to trading session.
4. Blocking (09:28 – 09:30 a.m.)	● Auction orders	- Order placement, change and cancellation shall be rejected.
	● Limit orders	- Same behavior as in Pre-order Matching

Appendix 4 – List of Error Code

Error Code	Minor Code	Error Message
1	10000	Order size is odd lot
-1	-10001	Account not specified
-2	-20000	Please specify Buy or Sell
-3	-30000	Invalid Symbol
-4	-40000	Please specify Quantity
-5	-50001	Insufficient position
	-50002	Insufficient close position
	-50003	Insufficient sell limit
	-50004	Insufficient stock position %.0n
-6	-60001	Cannot calculate credit amount for AO market order
	-60002	Insufficient trading balance/limit: need additional credit of %.0n
	-60003	Cannot %s from bank account
	-60004	Order over internet trading limit
-7	-70000	Price is not at spread level
	-70004	Price info is not available!
-8	-80001	Bid is over market by %.0n%%
	-80002	Bid is below market by %.0n%%
	-80003	Ask is below market by %.0n%%
	-80004	Ask is over market by %.0n%%
	-80005	Bid/Ask is not available
	-80006	Daily up limit check failed
	-80007	Decimal length check failed
	-80008	Can not do price checking
	-80009	Order price is over MaxQueue, please input the order again.
	-80010	Order price is over MaxTick, please input the order again
-9	-90001	Invalid Quantity
	-90002	Work quantity larger than order quantity.
	-90003	Order results in too many orders to the exchange. Please enter orders separately.
-10	-100001	Cannot save order:
	-100002	Cannot book trade:
-11	-110001	Account SELL Only for securities
	-110002	Account Close Only for futures/options
	-110003	Client account is suspended.
	-110004	User is suspended
	-110005	Security can be sold only.
	-110006	Order over upfront ratio.
	-110007	Account cannot trade in the exchange XXX
	-110008	Account cannot trade in the market YYY
	-110009	Insufficient approval limit: need additional amount of
	-110010	Insufficient per order approval limit: difference is

Error Code	Minor Code	Error Message
	-110011	Previous Change/Cancel still pending. Please RESET order
	-110012	Cannot change INAC order
	-110013	Exchange Order not replied
	-110015	Order quantity over max lots
	-110016	Order cannot be changed from your source
	-110017	Order cannot be cancelled from your source
	-110018	Client account is suspended via channel
	-110019	User is suspended via channel
	-110020	Exceed the maximum buying limit
	-110021	Account cannot trade this product
	-110022	Account is suspended for [EXCH] via channel [operator flag].
	-110023	User is suspended for [EXCH] via channel [operator flag].
	-110024	Account is suspended for product(%s: %s) via channel %s.
	-110025	User is suspended for product(%s: %s) via channel %s.
	-110026	Account is suspended for product(%s) via channel %s.
	-110027	User is suspended for product(%s) via channel %s.'
	-110028	User cannot trade this product(%s).
	-110029	Exceed the maximum sell volume limit.
	-110030	Total net-buy-limit exceeded. Required: %s
	-110031	Account cannot trade this symbol via channel %s
	-110032	Account need approval to trade this symbol via channel %s
	-110033	User is set inactive.
	-110034	Cross trade order is not allow to update via this channel.
	-110035	Invalid amount
	-110039	Submitted order is not allow to amend or cancel
	-110041	Invalid CCOG Investor ID.
	-110042	Order cannot be processed without Investor ID
	-110055	AO or ALO orders are not allow in CT.
	-110056	BCAN is invalid
-12	-120000	ORD-2:Order not outstanding
-14	-140000	Invalid exchange
-16	-160000	ORD-1:Change order failed. Trying to cancel order. Please inactivate order if it fails
-17	-170001	Stop price is over limit price
	-170002	Stop price is already at nominal
	-170003	Limit price is over stop price
-18	-180001	Order Amount over Transaction Limit
-19	-190000	Invalid Symbol
	-190001	Order passed last trading date.
	-190005	Reject order is not allowed in this trading session
-20	-200000	Lotsize of is zero
-21	-210002	Account Sell Only (ICL < 0).
-22	-220001	Market is closed
	-220002	Symbol status changed to day closed
-23	-230001	Cannot change AO order during market open

Error Code	Minor Code	Error Message
-24	-240001	Cannot approve SI instruction Order <order_no> not found Cannot approve non-inactive order Cannot approve Buy/Sell Transaction
-25	-250001	Cannot reduce work order quantity below <n>
-26	-260000	Only <n> order can wait for approval
-27	-270001	Cannot hold fund. Insufficient fund.
	-270002	Cannot hold fund. System failed.
	-270003	Cannot release fund
	-270004	Cannot release fund in change
	-270005	Cannot change hold fund
-28	-280001	cannot amend up
-29	-290001	Order amendment is blocked in current market status
-30	-300001	Order amount is over the warning limit.
	-300002	Add order met delisted symbol.
	-300003	Change order when symbol become delisted.
-33	-330001	Can not send Auction order at this market status: 1.
	-330002	AO buy order is not allowed.
	-330003	AO sell order is not allowed.
-42	-420000	Sell over day trade rule. Insufficient stock position %.0n
-99	-990000	Unknow extern checking error.
-9999	0	Sell over day trade rule. Insufficient stock position %.0n

Appendix 5 – Order State Change Matrices

A5.1 - Filled Order

Seq#	Action	Order Status (#5)	Order Qty (#4)	Order Price (#3)	Filled Qty (#34)	Avg Price (#42)	Error Code (#39)	Remark
1	Add		4000	59				
2		Reject					-7	If reject due to error "Price is not at spread level"
2		Pending	4000	59	0	$0.00	2	
3		Partial	4000	59	1000	$58.00	2	Execution of 1000
4		Partial	4000	59	3000	$58.17	2	Execution of 2000
5		Filled	4000	59	4000	$58.25	2	Execution of 1000

A5.2 - Cancel request issued for a zero-filled order

Seq#	Action	Order Status (#5)	Order Qty (#4)	Order Price (#3)	Filled Qty (#34)	Avg Price (#42)	Error Code (#39)	Remark
1	Add		1000	59				
2		Reject					-7	If reject due to error "Price is not at spread level"
2		Pending	1000	59	0	$0.00	2	
3	Cancel		1000	59				
4		Cancelled	1000	59	0	$0.00	2	

A5.3 - Cancel request issued for a partial-filled order

Seq#	Action	Order Status (#5)	Order Qty (#4)	Order Price (#3)	Filled Qty (#34)	Avg Price (#42)	Error Code (#39)	Remark
1	Add		4000	59				
2		Reject					-7	If reject due to error "Price is not at spread level"
2		Pending	4000	59	0	$0.00	2	
3		Partial	4000	59	1000	$58.00	2	Execution of 1000
4	Cancel		4000	59				
		Cancelled	4000	59	1000	$58.00	2	

A5.4 - Change order to complete

Seq#	Action	Order Status (#5)	Order Qty (#4)	Order Price (#3)	Filled Qty (#34)	Avg Price (#42)	Error Code (#39)	Remark
1	Add		2000	59				
2		Reject					-7	If reject due to error "Price is not at spread level"
2		Pending	2000	59	0	$0.00	2	
3	Change		3000	59				
4		Partial	3000	59	1000	$58.00	2	Execution of 1000
5	Change		1000	59				
		Filled	1000	59	1000	$58.00	2	

A5.5 - Change order with error

Seq#	Action	Order Status (#5)	Order Qty (#4)	Order Price (#3)	Filled Qty (#34)	Avg Price (#42)	Error Code (#39)	Remark
1	Add		2000	59				
2		Reject					-7	If reject due to error "Price is not at spread level"
2		Pending	2000	59	0	$0.00	2	
3	Change		3000	59				
4		Partial	3000	59	1000	$58.00	2	Execution of 1000
5		Partial	3000	59	2000	$58.13	2	Execution of 1000
6	Change		3000	58.35				
7		Reject					-7	If reject due to error "Price is not at spread level"
8	Change		4000	59				
9		Partial	4000	59	2000	$58.13	2	
10		Filled	4000	59	4000	$58.56	2	Execution of 2000

A5.6 - Change order resulting cancel

Seq#	Action	Order Status (#5)	Order Qty (#4)	Order Price (#3)	Filled Qty (#34)	Avg Price (#42)	Error Code (#39)	Remark
1	Add		2000	59				
2		Reject					-7	If reject due to error "Price is not at spread level"
2		Pending	2000	59	0	$0.00	2	
3	Change		3000	59				

4		Partial	3000	59	1000	$58.00	2	Execution of 1000
5		Partial	3000	59	2000	$58.13	2	Execution of 1000
6	Change		1000	59				
7		Cancelled	3000	59	2000	$58.13	2	Change order quantity < filled quantity resulting cancel

A5.7 - Change order resulting cancel

Seq#	Action	Order Status (#5)	Order Qty (#4)	Order Price (#3)	Filled Qty (#34)	Avg Price (#42)	Error Code (#39)	Remark
1	Add		2000	59				
2		Reject					-7	If reject due to error "Price is not at spread level"
2		Pending	2000	59	0	$0.00	2	
3	Change		3000	59				
4		Partial	3000	59	1000	$58.00	2	Execution of 1000
5		Partial	3000	59	2000	$58.13	2	Execution of 1000
6	Change		2000	59				
7		Cancelled	3000	59	2000	$58.13	2	Change order quantity = filled quantity resulting cancel

A5.8 - Change order with SUBERROR

Seq#	Action	Order Status (#5)	Order Qty (#4)	Order Price (#3)	Filled Qty (#34)	Avg Price (#42)	Error Code (#39)	Remark
1	Add		2000	59				
2		Reject					-7	If reject due to error "Price is not at spread level"
2		Pending	2000	59	0	$0.00	2	
3	Change		3000	59				
4		Partial	3000	59	1000	$58.00	2	Execution of 1000
5		Partial	3000	59	2000	$58.13	2	Execution of 1000
6	Change		3000	58.25				
7		Reject					-16	Reject due to order not outstanding
8		Filled	3000	59	3000	$58.25	2	Execution of 1000