LIFFE CONNECT® Conformance Guidelines

Preface

This document provides specific guidelines and instructions for application developers intending to write and conform to the LIFFE CONNECT® API.

It covers the requirements for all current operational LIFFE CONNECT® releases up to and including LIFFE CONNECT® release 8.0.

Target Audience

This document is aimed at developer and testing personnel of ISV and Member Developer firms who are intending to conform to a LIFFE CONNECT® API release.

Readers of this document should have a basic understanding of the LIFFE CONNECT® API.

Associated Documents

The following lists the associated documents, which either should be read in conjunction with this document or which provide other relevant information for the user:

API Developer Guidelines 8.0, www.liffe.com/access/developers/index.htm
API Reference Manual 8.0, www.liffe.com/access/developers/index.htm
API Functional Descriptions 8.0, www.liffe.com/access/developers/index.htm
API Changes 8.0, www.liffe.com/access/developers/index.htm
Developers Website http://liffedeveloperdiary.if5.com/

Table of Contents

1. Introduction

The Customer Technical Support Group (CTSG) safeguards the integrity of the LIFFE CONNECT® market place, of the exchange and its external clients. Euronext.liffe requires each trading member to use conformant software and demonstrate a level of technical and operational readiness before entering the market.

A full conformance test usually involves the following four components:

- Backwards Compatibility Testing
- Functionality Conformance
- Host Failure Conformance
- Volume Conformance.

2. Backwards Compatibility Testing

For ISVs and Member Developers who are already using conformed software from the LIFFE CONNECT® 7.1 release, it will be necessary to undertake a backwards compatibility test.

The basic requirement is for the ISV/Member Developer to ensure that the application that was developed to API 7.1 can continue to perform existing functionality whilst running on the CONNECT® 8.0 code base.

This is tested by ensuring the client application using API 7.1 connects to the CONNECT® 8.0 test environment and performs the following basic functions:

- Initialise API
- Logon
- Download of Standing Data
- Subscribe to market
- Submit Orders
- Revise Order
- Pull Order
- Receipt of Market messages.

3. Preliminary Conformance Testing

Any ISVs or Member Developers that are developing to the LIFFE CONNECT® API for the first time will be required to undertake a preliminary conformance test. The aim of this test is to ensure that the ISV or Member Developer can successfully implement the basic functions required when developing to the LIFFE CONNECT® API.

The following areas are covered during the test:

- Initialise API
- Logon
- Download of Standing Data
- Subscribe to market
- Submit Orders
- Revise Order
- Pull Order
- Receipt of Market messages.

Note: Unlike the functionality conformance test the preliminary conformance test does not need to be executed using a finished application.

4. Functionality Conformance

The following sections describe in detail each aspect of the functionality conformance and the completion criteria that the client application is expected to meet.

4.1 Test Structure

The primary aim of the functionality conformance test is to ensure that the client application does not behave in a way that will adversely affect the LIFFE CONNECT® Trading Host or any of its supporting systems.

This is achieved by the functionality conformance test. The test is conducted on a one-to-one basis over the telephone between a CTSG Analyst and the developer.

The CTSG Analyst will request the developer performs various functions using their client application.

The CTSG Analyst will verify that the client application interacts correctly with the API using monitoring tools on the test environment.

The conformance test will take place in the 'Own Market' test environment thus ensuring the test is not subject to any interference by a third party. A description of the 'Own market' and how the developer is expected to prepare for this is provided in Appendix A.

The functionality test itself consists of five modules which are designed to test functions in the five threads of the LIFFE CONNECT® API.

- Market Access
- Data
- Market Information
- Trading and Order Handling
- Market Control.

Each functionality conformance test is allocated a slot of a maximum of 3 hours. If the client application fails any part of the test they may have to re-take one or more of the modules. This is at the discretion of the CTSG Analyst conducting the test.

Note: During the functionality test the developer must conform the finished application that they intend to use in the live market.

4.2 Test Coverage

The developer will be required to demonstrate all aspects of the client application that interacts with the LIFFE CONNECT® API.

Note: It is not mandatory for the client application to support all functions of the API.

For ISVs the parent exchange will classify the level of functionality coverage according to a tiering system. Supporting more of the available functionality will result in the ISV being allocated a higher tier. This tiering information is then made available publicly so that members can decide on the most appropriate application to suit their business needs.

The requirements for reaching each specific tier are specified on the Developer Website - http://liffedeveloperdiary.if5.com.

There is no such classification for members wishing to develop bespoke applications for their own specific business needs.

4.2.1 Response Handler Functions – Return Values & Status Flags

It should be noted that the conformance test will not test the ability of the client application to correctly receive and interpret all possible values of each function callback.

For example, the conformance test may test a client application's ability to successfully submit an order but may not test whether the client application can identify that an order has been rejected due to it being outside price limits.

Developers should ensure that their code looks at **all** the parameters received within a Response Handler. Developers should also check that their code will cater for **all** possible values returned, in order to correctly handle the responses received. Not all of these will be testing during the conformance test. Some common failures to examine returned values and status flags are described in the Developer Guidelines document.

The CTSG Analyst will be able to assist, if requested, with testing any scenario on the test environment but it should be understood that not all possible scenarios will be tested during the conformance test.

4.3 Market Access

4.3.1 Initialise API

Once network connectivity has been established the developer can initialise the LIFFE CONNECT® API on a LIFFE CONNECT® gateway using either their client application or the testcon program.

To complete this test the developer must successfully initialise the API on the gateway and receive a return of LIFFE_SUCCESS_TEST. However, particular attention should be paid to the following fields in this function call:

- **Heartbeat Check**

  The value of the field `bDisableHeartBeatCheck` determines whether or not the client application will receive Trading Host Heartbeats indicating the trading host is 'alive'.

  The `dDisableHeatBeatCheck` field should be set to LIFFE_TRUE (i.e. heartbeats enabled) in the test environment to allow the developer to debug the client application.

  However, when the client application is implemented in the production environment this field must be set to LIFFE_FALSE. If this is not done the client application will not be able to connect.

- **Identifier String**

  The `IdentifierString` introduced in API version 6.0, should be used to clearly identify the software being used for that session. The contents of the field are to be specified by the client application and must uniquely identify the application being used. The information this field is to contain can be summarised as follows:

  - Solutions Provider i.e. ISV/MD Mnemonic.
  - Operating System i.e. Windows 2000, Solaris 8 etc.
  - Generic Application Version i.e. Workstation or Server version 2.0 etc.

  For example, an ISV with the mnemonic ABC and a Solaris 8 server-based solution could populate the field with the following string 'ABCS8svr2.0a'.

4.3.2 Prevention of Repeated Logon Failure

This functionality prevents a client application from repeatedly sending logon requests to the trading host when the application's ITM (key) is already logged on.

During the conformance test, the conformance analyst will simulate this by logging with an ITM belonging to the developer. The developer will then be asked to attempt to log on the application repeatedly.

The trading host will return the following values for the first (n) attempts:

`pcreturn.estatus`=LIFFE_USER_ALREADY_LOGGED_ON

`eReturn`=LIFFE_STILL_LOGGED_ON

On the (n+1) attempt the gateway will not forward the request onto the trading host. Instead the gateway will return the following values

`pcreturn.estatus`=LIFFE_TOO_MANY_LOGON_ATTEMPTS

`eReturn`=LIFFE_ERROR

In production the ITM would then be locked for a period of (n) minutes during which time no further log on attempts will be permitted.

To pass this test the client application must correctly log and interpret these messages for the end user.

Note: Applications which do not have an automatic logon function will not be able to complete this test.

4.3.3 Logon to the Trading Host

To pass this test the client application must be able to logon successfully to the trading host using `LiffeAccessLogon`.

4.3.4 Nominate Replacement Trader

This function is used to define the replacement trader, who takes over all non-GTC orders from the current trader.

To pass this test the client application must be able to successfully call `LiffeAccessNominate` with the `pszTrader` field containing an ITM specified by the CTSG Analyst.

4.3.5 Handover to Replacement Trader

This procedure is used to handover all outstanding non-GTC orders of the current trader to the pre-defined replacement trader and log the existing trader out.

To pass this test the application must be able to successfully call `LiffeAcessHandover` and receive an `OnAccessResponse` `eStatus`=LIFFE_STATUS_SUCCESS.

The developer should be aware of the other possible values of `eStatus` and correctly understand what these mean and how the application should handle them.

The CTSG Analyst will be able to assist the developer in testing other `eStatus` values but these will not form part of the conformance test.

4.4 Data

4.4.1 Download of Standing Data

This function is used to request all valid exchange codes, tradable contracts, contract month information and Automated Market References (AMR) for the exchanges and contracts nominated by the CTSG Analyst.

To pass this test the application must be able to successfully call `LiffeDataExchanges`, `LiffeDataContractInfo`, `LiffeDataContractMonths` and `LiffeDataMarkets`. In response to these, the application must be able to correctly receive and interpret `OnDataExchanges`, `OnDataContractInfo`, `OnDataContractMonths` and `OnDataMarkets`.

The results of these need to be viewed by the end user and recorded on the conformance script.

Note: The download of standing data will cover both derivative products and cash market products where appropriate.

4.4.2 Contract Notification

This response handler is transmitted when there is a change to the valid outright markets for a contract.

To pass this test, the application must be able to receive single or multiple new option strikes without re-subscription. This information will be transmitted via `OnDataCNotification` which will be triggered by the CTSG Analyst.

4.4.3 Strategy Notification

This function is used to notify the client application of strategy markets available for trading.

To pass this test, the application must be able to identify currently available strategies and receive notifications of newly created strategies without re-subscription. This will be achieved by calling `LiffeDataStrategies` and correctly interpreting existing strategy markets from `OnDataStrategies`.

New strategies will be delivered by `OnDataSNotification`. It will also be necessary to correctly receive and interpret these intra-day strategy updates.

4.4.4 Retrieve Orders

Using a Retrieve Orders call, the client application can retrieve current details of either:

- The trader's GTC orders in the Trading Host.
- The trader's non-GTC orders that were withdrawn at the point of a client application failure (on reconnection, during the same Trading Day).

Using the `LiffeMarketRetrieveOrders` call, the application will be required to correctly receive and interpret `OnMarketRetrieve`. The resulting order details will be recorded on the conformance script and will be verified by the CTSG Analyst.

4.4.5 Retrieve Fills

Using a Retrieve Fills function call the client application can retrieve details of all fills (executions) that have occurred during the current trading day.

Using the `LiffeTradeRetrieveFills` call, the client application will be required to correctly receive and interpret `OnMarketRetrieve`. The resulting fill list will be recorded on the conformance script and will be verified by the CTSG Analyst.

4.4.6 Create Market

This function is used to create a new strategy market and to obtain the corresponding Automated Market Reference (AMR).

All strategies are created from the buy perspective and the order in which the strategy legs must be specified is detailed in supporting documentation.

The application must be able to successfully create a number of futures and/or options strategies using the `LiffeDataCreateMarket` function call. `OnDataCreate` will provide the client application with the AMR. The Client application must be able to correctly receive and interpret this.

4.5 Market Information

4.5.1 Subscribe to Market

A trader must subscribe to a market before being able to receive any market information about it. There are 3 levels of subscription:

- **Commodity** – Receive market data for all outrights and strategies with a leg in, a given commodity.
- **Expiry** – Receive market data for all outrights and strategies with a leg in, a given expiry month for a given commodity.
- **Market** – Receive market data for a given outright or a specific strategy in a given commodity, specifying the Automated Market Reference (AMR).

To pass this test, the client application must make correct use of the `LiffeMarketSubscribe` call, recording both the `eLevel` parameter and the `eSubscribeFlag` subscription stream on the conformance script.

The developer may be asked to view and record the best bid and offer, market depth and implied prices for specified outrights and strategies.

Note: Subscriptions to both derivative and cash market products will be tested where appropriate.

If an application subscribes at Market Level the developer will need to e-mail the CTSG Analyst with details of the applications use of this call.

4.5.2 Market Depth

Due to the broadcast transmission of GTC market depth, when each market goes into Pre-Open, the `LiffeMarketDepth` request is now seen as redundant functionality by Euronext.liffe.

The application developer should ensure that if the market depth functionality is supported within their application, then it should be requested on a totally ad-hoc basis and at no time is the market depth request ‘triggered’ from a market state change, i.e. from Closed to Pre-Open.

This requirement will form part of the conformance criteria for entering the markets and hence should be strictly adhered to.

If a client application wishes to subscribe to depth it is recommended that the `LiffeMarketSubscribe` call is used for this utilising the appropriate values for `eSubscribeFlag`.

4.5.3 Order Book Update

This response handler is transmitted when there is a change to the explicitly quoted prices and/or volumes in the central order book.

The CTSG Analyst will make changes to the order book generating `OnMarketOrder` messages. To pass this test, the application must note and record details of the changes made to the order book.

Note: The Order Book Update test is only applicable to client applications that subscribe to the Depth subscription stream. See 4.5.1 Subscribe to Market.

4.5.4 Market Update

This response handler transmits the last traded price and volume, the best explicit bid and offer and the total traded volume. This is transmitted when there is a change to the best price or a trade in an outright market.

To pass this test, the application must correctly report updates that are transmitted via the `OnMarketUpdate` response handler.

Note: The developer should pay particular attention to the distinction between trade types using the `eTradeType` field. Specifically the client application should be able to handle all possible values of this field.

Problems have been experienced in the past when reporting block trades (`eTradeType=LIFFE_BLOCK`). Since these are executed outside the central order book the price can often be significantly different from the market price. This has in the past caused some client applications to trigger stop orders.

Note: The Market Update test is only applicable to client applications that subscribe to the Best Price subscription stream. See 4.5.1 Subscribe to Market.

4.5.4.1 Indicative Opening Prices

During Pre-Open the client application will be required to correctly receive and interpret the case where `OnMarketUpdate` is sent with `eExtent=LIFFE_INDICATIVE_OPENING` indicating indicative opening prices. The developer will be required to report the indicative opening prices delivered by the `OnMarketUpdate` callback.

Note: This test takes place when the market is in Pre-Open.

4.5.5 Strategy Market Update

This response handler transmits the last traded price and volume, the best explicit bid and offer and the total traded volume. This is transmitted when there is a change to the best price or a trade in a strategy market.

If the function is transmitted as a result of a trade, it also includes prices and volumes for each of the individual legs.

To pass this test, the application must correctly report updates that are transmitted via the `OnMarketStrategyUpdate` response handler.

4.5.6 Implied Market Information – Implied Out Prices

For a given outright market, this Response Handler Function will transmit implied prices and associated volumes to subscribed applications, when either:

1. An implied out buy/sell price can be calculated, and it is better than or equal to the best explicitly quoted price.

2. A previously transmitted implied buy/sell price or volume changes, or can no longer be implied.

3. When a Client Application subscribes to the Best Price stream for a market, and the implied buy/sell price for the market is better than or equal to the best explicitly quoted price.

Note: This function will not be transmitted when a fast market has been declared.

To pass this test, the application must correctly report implied updates that are transmitted via the `OnMarketImpliedUpdate` response handler.

4.5.7 Implied In Pricing

For specified strategy markets the Trading Host will calculate implied prices using explicit prices in the outright markets. These are called 'Implied In' prices.

The API will not report these prices to the client application. Instead it will be the responsibility of the client application to calculate the implied in prices using simple rules as specified by the strategy.

To pass this test, the application must correctly display implied in prices for all supported and relevant strategy markets. These will be verified by the CTSG Analyst.

4.6 Trading and Order Handling

4.6.1 Submit Orders

To pass this test, the application must be able to submit all supported order types for both outrights and strategies using the `LiffeTradeSubmitOrders` call.

The client application will be required to submit all supported order types by varying the `ePriceType` and `eTimeType` fields in the `pcOrderEntryList`.

Correct receipt of the response handler `OnTrade` or `OnTradeStrategy` will be tested to ensure the application reports the status of the order:

LIFFE_STATUS_SUCCESS

LIFFE_STATUS_ERROR

LIFFE_FAILED_TO_TRADE

LIFFE_PARTIAL_TRADE

LIFFE_ACCOUNT_CODE_INVALID

LIFFE_ORDER_OPEN_CLOSE_INVALID

LIFFE_STRATEGY_OPEN_CLOSE_INVALID

The settings of variables in the `pcOrderEntryList` structure will also be tested, specifically `chAccountCode`, `ePostingCode` and `chOpenClosedIndicator`.

The following tables list the valid values that can be submitted for the Open/Closed Indicator:

VALUE	OUTRIGHTS	TWO LEGGED STRATEGIES
O	Open	Open/Open
C	Close	Close/Close
<blank>	If no value is specified the indicator will default to Open	Defaults to Open/Open
X	Not applicable	The first leg is to be opened and the second leg is to be closed
Y	Not applicable	The first leg is to be closed and the second leg is to be opened

FOUR LEGGED STRATEGIES				
VALUE	**FIRST LEG**	**SECOND LEG**	**THIRD LEG**	**FOURTH LEG**
0	Open	Open	Open	Open
1	Open	Open	Open	Closed
2	Open	Open	Closed	Open
3	Open	Open	Closed	Closed
4	Open	Closed	Open	Open
5	Open	Closed	Open	Closed
6	Open	Closed	Closed	Open
7	Open	Closed	Closed	Closed
8	Closed	Open	Open	Open
9	Closed	Open	Open	Closed
A	Closed	Open	Closed	Open
B	Closed	Open	Closed	Closed
C	Closed	Closed	Open	Open
D	Closed	Closed	Open	Closed
E	Closed	Closed	Closed	Open
F	Closed	Closed	Closed	Closed

The following values that can be used for ePosting Code:

- LIFFE_POSTING_UNDEFINED
- LIFFE_POSTING_MANUAL

- LIFFE_POSTING_GIVE_UP
- LIFFE_POSTING_AUTOMATIC
- LIFFE_POSTING_AUTOMATIC_AND_ACCOUNT_AUTHORISATION
- LIFFE_POSTING_MANUAL_AND_ACCOUNT_AUTHORISATION

Note: Market on Open Orders (MOO) and Persistent Market on Open Orders (PMOO) will be tested in the Market Control Module of the conformance test.

4.6.2 Submit Market Making Orders

Market Making Orders (MMO) will allow Market Makers to submit batches of two sided quotes.

To pass this test, the application must be able to submit MMO orders using the `LiffeTradeSubmitMarketMakingOrders` call.

The client application will be required to correctly receive and interpret the `OnTradeMarketMakingOrdersSubmit` callback.

The developer should be aware that it is not possible to revise Market Making Orders using the `LiffeTradeReviseOrder` function.

Market Making Orders can effectively be revised by submitting new orders into markets with existing Market Making Orders. i.e. new MMOs replace existing MMOs.

Market Making Orders are only applicable to Amsterdam markets.

Note: The developer should be aware of restrictions to prevent overuse of this call which will form part of the volume conformance test.

4.6.3 Delta Protection

The Delta Protection functionality will assist in the management of Market Making Orders by tracking the net Delta Position of all trades. The LIFFE CONNECT® API allows the trader to specify what action should be taken if a pre-set Delta Limit is breached.

To pass this test, the client application must be able to set a Delta Limit for a particular contract using the `LiffeTradeSetDeltaProtection` function call.

The client application will be required to recognise all breach action types by varying the setting of the `eLimitBreachedAction`.

The CTSG Analyst will breach the limit a number of times and the client application will be required to correctly receive and interpret the responses.

The client application will additionally be required to adjust the current delta position using the `LiffeTradeAdjustDeltaProtection`.

The client application may also be required to obtain the delta protection status of a specified product using the function call `LiffeGetDeltaProtection`.

The Delta Protection function is only applicable to Amsterdam markets.

4.6.4 Stock Order Routing

The Stock Order Routing functionality allows members and traders who are not permitted to trade on the cash exchange to route stock orders through to members/trader who are able to trade on the cash market.

To pass this test, the client application will be required to submit Stock Order requests using the `LiffeTradeRouteStockOrderRequest` and route stock order data using `LiffeTradeRouteStockOrderData`.

Additionally the client application will be able to receive stock order notifications via `OnTradeRouteStockOrderRequest` and `OnTradeRouteStockOrderData`.

The Stock Order Routing function is only applicable to Amsterdam members.

4.6.5 Submit Ex-Pit Trades

The `LiffeTradeSubmitExPitTrade` call is used to submit Block, Basis, Against Actual, Guaranteed Cross and Prof Trade orders.

Ex-pit submissions will be tested for outrights, strategies and orders that contain an underlying leg. Block trades that are submitted outside market hours will also be tested.

To pass this test, the client application will have to correctly recognise cases where the ex-pit trade is both accepted and rejected by the CTSG Analyst.

The application must be able to correctly recognise the `OnTradeExPitTrade` callback with `eStatus`=LIFFE_STATUS_SUCCESS indicating the ex-pit trade has been accepted by the trading host. Note that this does mean the trade has been authorised by the exchange.

For ex-pit trades that are authorised by the CTSG Analyst, the application must be able to correctly handle the two `OnTrade` (or `OnTradeStrategy`) callbacks that are sent indicating that both the buy and the sell side of the ex-pit trade has been executed.

For ex-pit trades that are rejected, the application must be able to correctly recognise the `OnTradePull` callback with `eStatus`=LIFFE_OBSERVER_PULL_ORDER.

In order to support Prof Trades, a new `TradeType` (LIFFE_PROFESSIONAL) has been added to the API (API 8.0.3). The `LiffeTradeSubmitExPitTrades` function will be modified to accept orders of the new type. Where the trade type is set to LIFFE_PROFESSIONAL, the API allows the user to pass a single `LiffeExPitHalfTrade` struct to `LiffeTradeSubmitExPitTrades`. This will be sent to the Host for matching with the corresponding half trade submitted by the trader's designated counterparty. The `LiffeExPitHalfTrade` struct will be extended to include additional fields required for the processing of Prof trades.

On successful submission of the Prof trade intention the client application will receive an `OnTradeExPitTrade` callback with the status set to LIFFE_STATUS_SUCCESS. If the intention was submitted successfully, the appropriate `LiffeOrderId` (buy side or sell side) passed to the callback will contain the Order ID of the submitted intention. The other `LiffeOrderId` will be set to zero.

4.6.6 Submit Contingent Multiple Orders

A Contingent Multiple Order (CMO) is an order that contains two or more component orders. Trading of any component is contingent on being able to fully trade all components within the CMO. CMOs provide clients with the ability to trade across two separate contracts, therefore allowing traders to submit inter-contract spreads to LIFFE CONNECT®. They cannot be submitted during the Pre-Open period, as all the order components must exist in open markets.

CMOs are bound by the following:

- Maximum of 8 component orders.
- All component orders must be for outright orders, i.e. no strategies.
- Only one futures component permitted if any component order is for an option.

Note: It is not possible to submit a CMO for products on the equities Trading Host for LIFFE CONNECT® version 8.0.

Each component of a CMO can be of Limit or Market type. The CMO can be submitted as a mixture of buy and sell orders. The permitted product pairs are pre-defined by the Exchange.

The trades are executed in the same order as the components within the submitted CMO. All output messages accumulated during trading of the contingent order will be transmitted once trading of all the components is complete.

A single Order ID will be allocated to all of the components within a CMO.

To pass this test, the application must be able to submit orders for both outrights and strategies using the `LiffeTradeSubmitContingentOrder` call.

Correct receipt of the `OnTradeSubmit` response handler will be tested to ensure the application correctly handles the `eStatus` of the `pcOrderStatusList`:

LIFFE_STATUS_SUCCESS

LIFFE_STATUS_ERROR

LIFFE_FAILED_TO_TRADE

LIFFE_ACCOUNT_CODE_INVALID

Note: This functionality will not be available for Amsterdam markets.

4.6.7 Batch Submission of Orders

In addition to the validation undertaken on individual orders, multiple submission of orders are subject to the following validation:

- All orders within a Batch order submit must be for the same commodity.
- All orders within a Batch order submit must be Limit Orders. i.e. `ePriceType=`LIFFE_LIMIT.

- Multiple orders will be subject to crossing prevention, i.e. no two orders within a multiple order may cross.

Correct receipt of the `OnTradeSubmit` response handler will be tested to ensure the application can correctly handle the `eStatus` value in `pcOrderStatusList`. The following statuses will be covered:

LIFFE_STATUS_SUCCESS

LIFFE_STATUS_ERROR

LIFFE_PARTIAL_TRADE

LIFFE_ACCOUNT_CODE_INVALID.

In LIFFE CONNECT® 8.0 for Amsterdam products, the maximum batch size will be 1 order (therefore this functionality will not be available for Amsterdam markets).

4.6.8 Flex Order Submission

The trade submission and authorisation process is semi-automated, requiring manual intervention by an Exchange Official. Flex orders are applicable to London markets only.

The client application will be required to submit Flex orders for both outright and strategies using the `LiffeTradeSubmitFlexTrade` call and recognise the values of `eStatus` in `OnTradeFlexTrade`:

LIFFE_STATUS_SUCCESS – indicating the order has been accepted by the trading host.

LIFFE_FLEX_REJECTED – indicating the order has been rejected by the trading host.

The application must recognise the resulting `OnTrade` messages from Flex orders that are authorised by the CTSG Analyst.

For Flex trades that are rejected, the application must be able to correctly handle `eStatus`=LIFFE_OBSERVER_PULL_ORDER from the resulting `OnTradePull` response handler.

4.6.9 Revise Orders

When revising an order, the price, volume and/or the expiry can be amended.

Volume changes can be specified in three ways:

- The absolute working volume for the order after the amendment
- The change to the current working volume (as a positive or negative)
- A new value for the order's original volume.

In order to pass this test, the application must be able to successfully revise orders using the `LiffeTradeReviseOrders` call for each of the supported methods specified above.

In LIFFE CONNECT® 8.0 for Amsterdam products, the maximum batch size will be 1 order (therefore this functionality will not be available for Amsterdam markets).

4.6.10 Pull Orders

This function allows the client application to cancel specified working orders from the market.

To pass this test, the application must be able to pull orders specified by the CTSG Analyst using the function call `LiffeTradePullOrders` by all using each of the supported methods. There are three available:

- Individual orders for a given list of Order Id's (LIFFE_BY_ORDER)
- All orders in a futures or options contract (LIFFE_BY_COMMODITY)
- All orders in a given expiry month for a specific futures or options market (LIFFE_BY_EXPIRY_DATE).

For all other markets, the maximum batch size for cancellations will be 64 orders.

4.6.11 Receipt and Submission of RFQ

To pass this test, the client application will need to correctly receive and interpret an `OnTradeRFQ` generated by the CTSG Analyst.

Additionally the client application will be required to submit an RFQ using the `LiffeTradeSubmitRFQ` call.

4.7 Market Control

4.7.1 Market Mode

To pass this test, the application must respond to changes in market mode and recognise the change in status of a specified market and, if applicable, their working orders including Market On Open orders and Persistent Market on Open Orders.

The CTSG Analyst will manipulate the market state to generate `OnControlMode` messages. The client application will be required to receive and correctly recognise the value of `eMarketMode` and take appropriate action depending on the market state.

LIFFE_MARKET_PREOPEN - The Market is in Pre-Open mode.

LIFFE_MARKET_MODE_OPEN - The Market is in Open mode.

LIFFE_MARKET_MODE_PRECLOSED - The Market is in Pre-Closed mode.

LIFFE_MARKET_MODE_CLOSED - The market is in Closed mode. All non-GTC orders will be cancelled from the market.

LIFFE_MARKET_MODE_ENABLED –The contracts are available for trading (i.e. not suspended).

4.7.1.1 Market Disabled (Suspended)

The developer should pay particular attention to the situation where the market is disabled (sometimes referred to as suspended).

As stated in the API manual `eMarketMode` can be any bit-wise combination of the market states listed above. The LIFFE_MARKET_MODE_ENABLED enumeration is used to identify whether the market is enabled or disabled. If LIFFE_MARKET_MODE_ENABLED is present in `eMarketMode` (i.e. Enabled=TRUE) then the market is enabled.

If LIFFE_MARKET_MODE_ENABLED is not present in `eMarketMode` (i.e. Enabled=FALSE) then the market is disabled or suspended.

When the market is disabled the Trading Host cancels all active orders from the market including GTC's. The client application must be able to correctly recognise when the market is disabled and cancel all orders from their local order book. The trading host will not send cancel notifications for any active orders.

Whilst the contract is in Pre-Open the application must be able to recognise indicative opening prices for both futures and options and submit MOO and PMOO orders using the `LiffeTradeSubmitOrders` call with a `LiffeTimeType` of LIFFE_MARKET_ON_OPEN and LIFFE_PERSISTENT_MARKET_ON_OPEN respectively.

4.7.2 Receipt of Market Messages

The application should be able to confirm the receipt of a message sent via the `OnControlMessage` response handler.

4.7.3 Forced Logout

The application must be able to differentiate between a user log off and a forced log out. This is specified in the `OnControlLogout` response handler with an `eStatus` of LIFFE_OBSERVER_LOGOUT.

4.7.4 Trader Exchange Suspend

The LIFFE CONNECT® 8.0 release now permits a single ITM to access all markets.

Market Control officials can subsequently prevent ITMs from accessing a particular exchange by suspending them from the given exchange.

The CTSG Analyst will suspend the ITM that client application is using from a selected exchange. The client application will be required to correctly receive and interpret the `OnControlTraderExchangeSuspend` callback.

4.8 Host and Gateway Failure

The Host and Gateway Failure Conformance test is aimed at ensuring the client application can correctly recognise and react to various failure scenarios that could potentially occur in the live market.

The Host failure test lasts 1 hour and consists of a series of failure events. At each stage the client application must be able to react appropriately.

The schedule of events, together with appropriate timings is given to the external developer prior to the test. The CTSG Analyst then monitors the progress remotely.

Note: The host failure conformance test can only take place on applications that have successfully passed functionality conformance.

4.8.1 Trading Host Fails

Users will receive an `OnControlStatus` message, set with the `bAvailable` field set to LIFFE_FALSE, for all contract listed on the trading host, to indicate that the contracts are unavailable for trading.

Users will remain connected as the Sentinel host is still running.

All non-GTC orders will be discarded whilst all GTC orders will be removed from the order book but retained by the trading host. It will be possible to retrieve these orders when the trading host is re-started.

However, it is the responsibility of the client application to ensure that the local order book is cleared at the time of failure.

4.8.2 Trading Host Back Up

Users will receive an `OnControlStatus` message, set with the `bAvailable` field set to LIFFE_TRUE, for all contracts listed on the trading host, to indicate that the contracts are available for trading. The client application can re-subscribe to the contracts at this point.

At this point the client application should also retrieve their GTC orders that were active at the point of trading host failure. The client application will also be required to retrieve their fill information for the current trading day.

The market state will moved from Closed to Pre-Open and finally Open.

4.8.3 Sentinel Host Fails

This step is invisible to the user. Trading can continue as normal.

4.8.4 Trading Host Fails

As the Sentinel Host has already failed, the user will lose connection once the trading host fails. The client application will receive 2 `OnAccessResponse` messages with a status of LIFFE_GATEWAY_CONNECTION_ERROR and LIFFE_EXCHANGE_CONNECTION_ERROR.

All non-GTC orders will be discarded whilst all GTC orders will be removed from the order book but retained by the trading host. It is the responsibility of the client application to ensure their local order book has been cleared.

4.8.5 Trading Host Back Up

The CTSG Analyst will inform the developer when the Trading Host has been re-started. The application can then be restarted and it will be possible to log back on. Users should re-subscribed and retrieve any GTC orders that were present at the time of the Trading Host Failure.

4.8.6 Pre-Open

Users should submit Market On Open orders at this point. Indicative Opening Prices will also be generated by the CTSG Analyst. The client application will be required to display these and the prices should be communicated to the CTSG Analyst.

4.8.7 Market Open

All contracts are fully open and Market On Open orders will have traded out.

Settlement prices for futures and options will also be generated at this point. The client application will be required to display these and the prices should be communicated to the CTSG Analyst.

4.8.8 Gateway Failure

The trading host will fail and users will receive an `OnAccessResponse` message with a status of LIFFE_GATEWAY_CONNECTION_ERROR.

The client application will be required to correctly recognise this.

5. Volume Conformance Testing

Volume conformance ensures that the client application can perform normal trading functions whilst the market is subjected to a high volume of trading activity.

Specifically the client application must satisfy the following criteria:

- Connect to Gateway and logon to the Trading Host
- Subscribe to Market
- Submitting Orders (including over-use of inbound throttle to ensure error messages are correctly handled)
- Revising Orders
- Pulling (Cancelling) Orders.
- Remain logged on for a continuous period of at least 30 minutes.

Each of these functions will be recorded on the conformance script and verified by the CTSG Analyst. The performance of the application will be continually monitored by the CTSG Analyst over the period of the test. Specific post-test analysis will be carried out the following day.

Volume conformance can only take place on an application that has successfully passed functionality conformance.

For LIFFE CONNECT® Release 8.0 volume conformance will take place over production infrastructure (gateways) to ensure that the test mirrors live conditions as closely as possible.

However, the specific infrastructure requirements for volume conformance will vary depending on the type of developer and the access solution chosen by the Member Developer or ISV. The following section summarises what infrastructure will be required in each place.

5.1.1 ISVs

A proportion of ISVs will have new LIFFE CONNECT® 8.0 gateways installed at their site (remote gateways) over which they will connect their applications and complete volume conformance test.

The remaining ISVs will need to enlist the assistance of a ‘friendly member’ and use their production gateways to complete the volume conformance test.

For ISVs that have a connection for live pricing this test can only start at 22:00 London time. If however they agree to have their production gateway taken down, they can start conformance at the usual time (currently 18:30 London time)

For ISVs that need to use a friendly member they will have to start the test at 22:00 London time. If however the friendly member has a completely new installation, they can start conformance at the usual time (currently 18:30 London time)

5.1.2 Member Developers

Member Developers who are already trading in production will have their existing production gateways re-pointed to the test environment and complete volume conformance using their existing infrastructure. This test can only start at 22:00 London time.

New Member Developers who have a new installation can start conformance at the usual time (currently 18:30 London time).

5.1.3 Light Access Solutions

Members who are making use of the light access solutions such as Internet VPN and 128K leased line will connect to gateways located in their nearest PoP (Amsterdam, Paris, Brussels or London).

5.2 Volume Profiles

There will be volume profile options available for Amsterdam volume conformance. ISVs and MDs that do not intend to support the Amsterdam market will be subjected to a volume conformance using the existing profiles. These profiles are not part of this document.

Volume will be allocated at rates of:

- Euronext Markets including London, Paris and Brussels - 10% of Volume messages
- Amsterdam Markets - 90% of Volume Messages, of which 60% will allocated to the AEX contract.

5.2.1 Recommended Subscriptions for Light Access Users

There are a number of possible combinations of contract subscriptions across a light access solution. These can be determined from the average number of IOMs in each liquidity group below. Each liquidity group is specified by a band of daily IOMs into which the contracts fall.

	AEX	**AEX LIGHT**
Market Percentage	57%	2%
“Tokens” per contract	900	30
“Tokens” with no depth	325	12

The number of “tokens” for each group of contracts can be used to calculate the range of contracts that can be traded across a light access solution.

The table below gives a view of the available “baskets” of contracts available over a light access solution. The calculations for contract subscriptions have been based on a token allocation approach. A light access solution can subscribe to an allocation of **100** tokens. The total contract subscription tokens can be calculated by aggregating the token

allocation of each of the contracts based on liquidity group. In this case over-subscription can be defined as the sum of the contract tokens exceeding the total subscription allocation.

	LIQUIDITY GROUP			
	Super/High	**Medium**	**Low**	**Other**
No. of Classes	7	13	17	12
Market Percentage	21%	15.5%	1.5%	0.5%
“Tokens” per contract	50	20	1.5	0.5
“Tokens” with no depth	20	8	0.5	0

This will allow the following combinations with full depth.

For example:

Futures Trader

- All Futures contracts

Options Trader

- 2 “super” or “high” liquid contracts, or
- 1 “super” or “high” liquid contract, and 3 “medium” liquid contracts, or
- 5 “medium” liquid contracts

Futures and AEX Trader

- All Futures and a month (or less in the front quarterly) in AEX Option
- All other combinations can be calculated using token allocation.
- Medium liquidity includes the AEX-light Option. It is assumed that all Futures and “Other” liquid contracts can be subscribed to in addition to those shown in each row above.

Appendix A. Use of the Own Market

Functionality conformance tests for LIFFE CONNECT® Release 8.0 will take place in the 'Own Market'. This section describes what the own market is and what will be required of the client application in order to complete conformance.

Each Member Developer and ISV is allocated a set of products for their own exclusive use. This product set will typically contain a product for each of the different product groups on the exchange. For instance each product set in the own market for Euronext.liffe contains the following contracts:

- London Short Sterling Future
- London Short Sterling Option
- London FTSE 100 Index Future
- London FTSE 100 Index Option

The fundamental difference between the own market and the dummy market is the naming convention of the contracts.

Due to the fact that each external Developer has their own set of products the standing data lists duplicate products (i.e. one for each external developer).

Each of these duplicate contracts are treated as separate products and are distinguished from one another by the `PhysicalCommodityCode` field.

The following example shows the call back from `LiffeDataContractInfo` from the Environment 11 own market. This shows a number of instances of the Short Sterling Futures contract.

```
pcLiffeContractInfoList[11].chExchangeCode='L'

pcLiffeContractInfoList[11].szPhysicalCommodityCode="LN1"

pcLiffeContractInfoList[11].chGenericContract='O'

pcLiffeContractInfoList[11].szCommodityDescription="THREE MONTH
STERLING

pcLiffeContractInfoList[11].szProductGroup="   "

pcLiffeContractInfoList[11].nDefaultLotSize=1

pcLiffeContractInfoList[11].szContractCurrency="GBP"

pcLiffeContractInfoList[11].fTickValue=1.2500

pcLiffeContractInfoList[11].chCalculationType='I'

pcLiffeContractInfoList[11].chExerciseType='A'

pcLiffeContractInfoList[11].nStrikeScalingFactor=1
```

```
pcLiffeContractInfoList[11].chTradingType='T'

pcLiffeContractInfoList[12].chExchangeCode='L'

pcLiffeContractInfoList[12].szPhysicalCommodityCode="LN2"

pcLiffeContractInfoList[12].chGenericContract='O'

pcLiffeContractInfoList[12].szCommodityDescription="THREE MONTH
STERLING

pcLiffeContractInfoList[12].szProductGroup="   "

pcLiffeContractInfoList[12].nDefaultLotSize=1

pcLiffeContractInfoList[12].szContractCurrency="GBP"

pcLiffeContractInfoList[12].fTickValue=1.2500

pcLiffeContractInfoList[12].chCalculationType='I'

pcLiffeContractInfoList[12].chExerciseType='A'

pcLiffeContractInfoList[12].nStrikeScalingFactor=1

pcLiffeContractInfoList[12].chTradingType='T'

pcLiffeContractInfoList[13].chExchangeCode='L'

pcLiffeContractInfoList[13].szPhysicalCommodityCode="LN3"

pcLiffeContractInfoList[13].chGenericContract='O'

pcLiffeContractInfoList[13].szCommodityDescription="THREE MONTH
STERLING

pcLiffeContractInfoList[13].szProductGroup="   "

pcLiffeContractInfoList[13].nDefaultLotSize=1

pcLiffeContractInfoList[13].szContractCurrency="GBP"

pcLiffeContractInfoList[13].fTickValue=1.2500

pcLiffeContractInfoList[13].chCalculationType='I'

pcLiffeContractInfoList[13].chExerciseType='A'

pcLiffeContractInfoList[13].nStrikeScalingFactor=1

pcLiffeContractInfoList[13].chTradingType='T'
```

Each instance is distinguished by the Physical commodity code. Specifically the last two characters of the Physical commodity code are used to allocate the product to a specific external developer.

This is done by configuring the trading key to permit access to a single set of products on the own market.

For instance the trading key issued to External Developer A will be given exclusive access to all products will a physical commodity code ending in ‘N1’.

Hence in order to subscribe to Short Sterling the client application will have to submit a `LiffeMarketSubscribe` API call with physical commodity code “LN1” (using the above example).

This information will be communicated to all external developers prior to the conformance test.

In order to pass conformance for the LIFFE CONNECT® 8.0 release all client applications will need to be able to use the own market.

Appendix B. Contacting CTSG

To book a CTCI conformance test or for further information please contact CTSG:

- Telephone: +44(0)207 379 2983
- Telephone (for members located in the Netherlands): 0800 023 0932
- Email: ctsg@liffe.com

Appendix C. Conformance Key Dates

EVENT	DESCRIPTION	DATE
ISV/Member Developer Functionality Conformance Testing	Conformance testing to ensure all applications are conformed to LIFFE CONNECT® API 8.0.1	07 June – 30 July 2004
LIFFE CONNECT® API 8.0.2 available	Updated release made available	29 June 2004
ISV/Member Developer Volume Testing	Volume Conformance for ISV and Member Developers for LIFFE CONNECT® API 8.0.2	01 – 30 July 2004
LIFFE CONNECT® API 8.0.3 available	Updated release made available	13 September 2004
Backwards Compatibility Conformance against 8.0.3	Ensures Backwards Compatibility of LIFFE CONNECT® API 7.1 with 8.0.3 Host	13 September – 24 September 2004
ISV/Member Developer testing with API 8.0.3	Development and test period	13 September – 08 October 2004
API 8.0.3 Mini-functional conformance	Mini-functional conformance period	20 September – 15 October 2004

Appendix D. API Releases

API VERSION	DESCRIPTION
8.0.1	All existing Connect 7.1 and new Connect 8.0 functionality. (See API 8.0.1 Changes document for full details)
8.0.2	Inbound Order Throttle (See Order Input Control document for full details)
8.0.3	Prof Trades Account Authorisation Changes to Open/Closed Indicators (See API 8.0.3 Changes document for full details)

Euronext, LIFFE and Euronext.liffe

Euronext was formed by the merger of the Amsterdam, Brussels and Paris stock and derivatives markets exchanges in September 2000. The Euronext group has continued to grow through its merger with BVLP (the Portuguese stock and options exchange) and the acquisition of LIFFE (the London International Financial Futures and Options Exchange). The derivatives businesses of Euronext and LIFFE are being combined under the Euronext.liffe umbrella, and all of Euronext.liffe's derivatives marketplaces will go through a phased migration to LIFFE CONNECT®, the world's most advanced derivatives trading platform.

Euronext

Amsterdam
P.O. Box 19163
1000 GD Amsterdam
The Netherlands
phone +31 (0)20 550 6315
fax +31 (0)20 550 4975

Brussels
Palais de la Bourse/Beurspaleis
Place de la Bourse/Beursplein
1000 Brussels
Belgium
phone +32 (0)2 509 12 11
fax +32 (0)2 509 12 12

Lisbon
Praça Duque de Saldanha, 1-5°A
1050-094 Lisbon
Portugal
phone +351 (0)21 790 00 00
fax +351 (0)21 795 20 21

London
Cannon Bridge House
1 Cousin Lane
London EC4R 3XX
United Kingdom
phone +44 (0)20 7623 0444
fax +44 (0)20 7588 3624

Paris
39, rue Cambon
75039 Paris cedex 01
France
phone +33 (0)1 49 27 10 00
fax +33 (0)1 49 27 14 33

www.euronext.com