eSpeed API C/C++ Application Programming Interface

EXCHANGE SUPPORT

System Version: 1.4.13

Revision Date: April 24, 2006

1 Overview

This document presents the mechanism for submitting orders to and receiving trade confirmations from exchanges such as CBOT and Eurex through the eSpeed API. Full eSpeed API documentation is available in the *C/C++ Application Program Interface: eSpeedAPI Reference Guide*.

2 Customer Support

Customer Support is available between 7:00am and midnight U.S. Eastern time, Monday through Friday, excluding holidays. When calling, please assist us by being ready with your product and account information.

The eSpeed Customer Support group has a series of phone numbers and e–mail addresses to meet various user needs. They are as follows:

- If you are a Customer with questions about possible trading scenarios or specific trading features (e.g. price improvement) please contact your account representative.

- If you are a Customer experiencing technical difficulty, have questions on how to use the system, please call or write:

  eSpeed Call Center (US)   —   (+1) (212) 610-2300 or   support@espeed.com
  eSpeed Help Desk (Europe)   —   (0)20-7894–8600 or support@espeed.co.uk

- If you are a Customer or Salesperson requesting information on how to make changes to your electronic account, or require a new access account, please call or write:

  eSpeed Customer Access (US)   —   (212) 610-2300   or   customeraccess@espeed.com
  eSpeed Customer Access (Europe)   —   (0)20-7894-8886 or customeraccess@espeed.co.uk

- If you are a Customer with questions regarding API development issues, downloading the latest version of the SDK or JNI, or testing a trade feed or market–making application, please call (between the hours of 9:00am and 6:00pm EST) or write:

  eSpeed Customer Integration — (+1) (212) 610-3560 or customerintegration@espeed.com

- If you are a Customer with questions regarding specific trades, verification of a trade, or are experiencing delivery problems with a trade, please call:

  eSpeed Trade Support (US)   —   (212) 610-2300
  eSpeed Trade Support (Europe)   —   (0)20-7894-8600

3 Authentication

When connecting to an eSpeed trading system that presents an interface to an external exchange (e.g. CBOT) the accepted connection shall indicate to the application that the connection the connection is to an

exchange rather than an eSpeed hosted business. This shall be delivered to the application through the trade control flags in the CFETI_CONNECT_INFO data structure delivered to the application trading system session callback when the connection is accepted. Exchanges usually only accept buy or sell orders rather than bids or offers; typically the CFETI_BID_DISABLED and CFETI_ASK_DISABLED flags shall also be set in the trade control flags.

The following trade control flags are defined and relevant to applications written to access exchanges through eSpeed.

Flag (CFETI_)	Description
CFETI_BID_DISABLED	Applications shall not be permitted to submit markets where the command is CFETI_MARKET_BID.
CFETI_ASK_DISABLED	Applications shall not be permitted to submit markets where the command is CFETI_MARKET_ASK.
CFETI_BUY_DISABLED	Applications shall not be permitted to submit orders where the command is CFETI_ORDER_BUY.
CFETI_SELL_DISABLED	Applications shall not be permitted to submit orders where the command is CFETI_ORDER_SELL.
CFETI_EXCHANGE_INTERFACE	Trading system provides an interface to an external exchange.

After successful connection to an exchange a notification CFETI_SET_EXCHANGE_INFO shall additionally be delivered to the trading system connection callback. The command data delivered shall be a pointer to a CFETI_EXCHANGE_INFO_DESC data structure, defined as follows.

```
typedef struct CFETI_EXCHANGE_INFO_DESC CFETI_EXCHANGE_INFO_DESC
struct CFETI_EXCHANGE_INFO_DESC {
    unsigned int     exchangeId;        /**< Enumerated exchange identifier */
    unsigned int     exchangePermissions; /**< Permissions bit-mask */
};
```

The following enumerations are provided for the exchange identifier.

Flag (CFETI_EXCHANGE_ID_)	Description
CBOT_ACE	Chicago Board of Trade, A/C/E (EasyScreen v49)
ECBOT_V1	Chicago Board of Trade, eCBOT (EasyScreen v49+)
ECBOT_V2	Chicago Board of Trade, eCBOT (EasyScreen v71)
EUREX_ACE	Eurex, A/C/E
LIFFE	LIFFE
EUREX_US	Eurex US
CME	Chicago Mercantile Exchange

The exchangePermissions bit-mask is used to identify additional exchange-specific functionality where it is available. The following values are currently provided that may be included in the bit-mask.

Constant (CFETI_)	Description
EXCHANGE_ACCOUNT_ADMIN	The administration of accounts is centrally managed in this exchange implementation. Mechanisms are provided to allow client applications to retrieve the list of client accounts available and to obtain the details of specific accounts.

4   Account Administration

If the exchange permissions bit-mask delivered to the client application in the command data when the connection callback is invoked with the command CFETI_SET_EXCHANGE_INFO includes the bit CFETI_EXCHANGE_ACCOUNT_ADMIN then the client application can opt to retrieve account details from the trading system interface.

The following commands can be issued using the CFETIPostMessage interface.

Command (CFETC_EXCHANGE_)	Description
LIST_ACCOUNTS	This command is used to request that the details of all available accounts should be delivered to the requestor. The command data is ignored when this command is issued.
GET_ACCOUNT	This command is used to request that the detail of a specific account be delivered to the requestor. If this command is issued the command data shall be a pointer to a CFETI_EXCHANGE_ACCOUNT_INFO_DESC data structure in which account name field has been populated.

The following commands will be issued to the trading system connection callback in response to the issued request.

Constant (CFETI_EXCHANGE_)	Description
LIST_ACCOUNTS_ACCEPTED	Request to list accounts is accepted and results are returned in the message. The command data delivered to the application shall be a pointer to a CFETI_EXCHANGE_ACCOUNT_LIST_DESC data structure.
LIST_ACCOUNTS_REJECTED	Request to list accounts is rejected. The command data structure shall not be valid when this command is delivered to the application.
GET_ACCOUNT_ACCEPTED	Request to get the details of the specified account was accepted and the results are returned in the message. The command data delivered to the application shall be a pointer to a CFETI_EXCHANGE_ACCOUNT_INFO_DESC data structure.
GET_ACCOUNT_REJECTED	Request the get the detail of the specified account was rejected. The command data delivered to the application shall be a pointer to a CFETI_EXCHANGE_ACCOUNT_INFO_DESC data structure. The only valid field in the account info data structure shall be the account name.

If the request to list accounts or to get the detail of a specific account is rejected, then one of the following values will indicate the reason for the rejection of the request.

Constant (CFETI_EXCHANGE_)	Description

NO_ACCOUNTS	No list of accounts is available.
ACCOUNT_NOT_FOUND	The account specified in a get request does not exist.
REQUEST_IN_PROGRESS	A request was made to retrieve the list of accounts for the user on a specific exchange but there is an existing request already in progress.

The following data structure shall be delivered to applications written to the eSpeed API when the command is CFETI_EXCHANGE_LIST_ACCOUNTS_ACCEPTED.

```
typedef struct CFETI_EXCHANGE_ACCOUNT_LIST_DESC CFETI_EXCHANGE_ACCOUNT_LIST_DESC;
struct CFETI_EXCHANGE_ACCOUNT_LIST_DESC {
     unsigned int      numAccounts; /**< Number of accounts */
     const CFETI_EXCHANGE_ACCOUNT_INFO_DESC* accounts; /**< List of accounts */
};
```

The following data structure shall be delivered to applications written to the eSpeed API for accepted or rejected account get requests. The same data structure is also used to request the account details; in this case only the accountName field is expected to be valid.

```
typedef struct CFETI_EXCHANGE_ACCOUNT_INFO_DESC CFETI_EXCHANGE_ACCOUNT_INFO_DESC;
struct CFETI_EXCHANGE_ACCOUNT_INFO_DESC {
     const char*       accountName; /**< Name given to the account */
     unsigned int      accountVersion; /**< Account version */
     CFETI_EXT_PROPERTY_ORDERINFO_DESC; /**< Account properties */
};
```

5   Order Structure

The fields `orderInfo` and `orderInfoType` in the order data structure are used to communicate business specific order attributes. The `orderInfoType` field is used to determine what type of information is stored in the order info. The `orderInfo` field should be treated as invalid if the `orderInfoType` field is zero.

For an order to an exchange the `orderInfoType` field shall have the value `CFETI_ORDERINFO_EXTENDED_PROPERTIES`.

The `orderInfo` field is then a pointer to a `CFETI_EXT_PROPERTY_ORDERINFO_DESC` data structure. This is a structure containing an integer to specify the number of properties attached to the order and then an array of these properties.

```
    /**
     * CFETI_EXT_PROPERTY_DESC: Preference struct for exchanges
     */
    typedef struct CFETI_EXT_PROPERTY_DESC CFETI_EXT_PROPERTY_DESC;
    struct CFETI_EXT_PROPERTY_DESC {
        unsigned int propertyId;
        unsigned int propertyType;
        CFETI_FIELD_VALUE propertyValue;
    };

    /**
     * CFETI_EXT_PROPERTY_ORDERINFO_DESC: Order info struct for exchanges
     */
    typedef struct CFETI_EXT_PROPERTY_ORDERINFO_DESC CFETI_EXT_PROPERTY_ORDERINFO_DESC;
    struct CFETI_EXT_PROPERTY_ORDERINFO_DESC {
        unsigned int numProperties;
```

```
        CFETI_EXT_PROPERTY_DESC* propertyList;
    };
```

For each supported exchange, a description of each exchange-specific extended order property, and the range of defined available values where appropriate, are listed below.

5.1   CBOT

Property (CFETI_EXT_PROPERTY_)	Data Type	Description
ORDER_TYPE	Uint32	The order type property defines the exchange order type of the orders. The value of this property shall have one of the following enumerated values (each defined constant shall have a prefix CFETI_EXT_PROPERTY_): ▪ MARKETORDER ▪ LIMITORDER ▪ STOPORDER ▪ STOPLIMITORDER
ORDER_TIME_TYPE		The order time type defines the time restriction that shall be applied to the order. The value of this property shall have one of the following enumerated values (prefix CFETI_EXT_PROPERTY_): ▪ ORDER_GOOD_FOR_DAY ▪ ORDER_GOOD_TILL_CANCEL The default order restriction that shall be applied if none is specified is that the order shall be good for the day on which it was submitted.
ORDER_RESTRICTION		The order restriction is used to place further restriction upon the order. The following order restriction enumerations are defined (prefix CFETI_EXT_PROPERTY_) ▪ ORDER_IMMEDIATE_OR_CANCEL ▪ ORDER_COMPLETE_VOLUME
ORDER_OPENPOSITION	Uint32	Whether or not the order is to open or close a position. This attribute is not delivered in trade confirmations. If the order is to open a position the field shall have a zero value or else if it is to close a position the value shall be non-zero. This is a required attribute.
MINIMUM_VOLUME	Decimal	Minimum volume attached to the order. If the value is not specified then no minimum volume is assumed. This attribute is not delivered in trade confirmations.

ACCOUNT_CODE	String	String used by to define properties of the account under which the order is submitted to the exchange. The following account code strings are valid: *space*　No give-up specified G　Give-up to single firm A　Give-up to multiple firms M　Market maker
LABEL	String	A free-format text label that can be attached to the order and returned in subsequent messages for that order. Note that some exchanges may impose an upper bound on the length of the string that can be used. This limit is not enforced in the eSpeed implementation.
CTI_CODE	Uint32	This order property is specific to CBOT and defines the Customer Type Indicator. The following customer type indicator enumerations are defined (prefix CFETI_EXT_PROPERTY_CTI_CODE_) ▪ OWN_ACCOUNT ▪ HOUSE_PROPRIETARY ▪ OTHER_MEMBER ▪ CUSTOMER_ACCOUNT
ORDER_ORIGIN	Uint32	This order property is specific to CBOT and defines the order origin. The following order origin enumerations are defined (prefix CFETI_EXT_PROPERTY_ORDER_ORIGIN_). ▪ CUSTOMER ▪ NON_CUSTOMER ▪ HOUSE_FLOOR_TRADER ▪ FLOOR_TRADER
CUSTOMER_ACCOUNT	String	A customer account name that can be attached to the order. Note that some exchanges may impose an upper bound on the length of the string that can be used. This limit is not enforced in the eSpeed implementation.
EXCHANGE_TRADE_REF	String	The exchange trade reference is the identifier given to an order placed on the exchange. It is delivered in trade confirmations for exchange orders by eSpeed in addition to the eSpeed trade identifier assigned to each trade to uniquely define the trade on eSpeed. It remains unchanged while an order remains on the exchange, no matter how many partial fills are made against it. If the order is modified (in price and/or volume), however, it does change, as the exchange regards this as a new order.
TRS_ORDER_ID	String	The exchange TRS Order ID is assigned to each partial fill (trade) made against an order placed on the exchange.
PRICE_FORMAT_CODE	Uint32	Enumerated value to identify the underlying format of the traded price. The possible values are: ▪ CFETI_PRICETYPE_CBOT_32ND

		▪ CFETI_PRICETYPE_CBOT_HALF_32ND ▪ CFETI_PRICETYPE_CBOT_QTR_32ND
CLIENT_ID	String	This field is used to deliver the exchange trader identifier of the user in trade confirmations sent to trade feeds.
EXCHANGE_ACCOUNT_NAME	String	The client application can opt to specify the account name instead of the extended properties that define the account. In this case the eSpeed trading system interface shall insert any account attributes that are not specified in the order before submitting the order to the exchange. This facility is available only where the exchange interface supports the account management function.

Trades confirmed for CBOT populate the instrumentIdType field in the CFETI_ORDER_DESC data structure with the value CFETI_INSTRUMENT_ID_ISIN and the instrumentId field with the Automated Market Reference. The interpretation of the elements of the AMR are as follows for outright future or option contracts and strategies:

1. Outright Future or Option contract

    Example: F F ZN 05 06 00000 F

    F   Exchange, 1 character (F = Financials)
    F   Instrument type, 1 character (F = future)
    ZN   Symbol of the product on the exchange. Two or more characters.
    05   Expiry year of contract
    06   Expiry month of contract
    00000   Strike price (if option)
    F   Instrument identifier, 1 character (F = future contract outright, C = call, P = put)

2. Strategies

    Example: F F ZB 02 E 000037 S

    F   Exchange, 1 character (F = Financials)
    F   Instrument type of leg components, 1 character (F = future)
    ZB   Symbol of the let component on the exchange. Two or more characters.
    02   Number of legs, 2 characters
    E   Strategy type, 1 character (E = calendar spread, B = butterfly spread, W = condor, M = strip, O = pack, Y = bundle, Z = reduced tick calendar
    000037   Not specified by the exchange. 6 characters.
    S   Instrument identifier, 1 character (S = strategy)

5.2   Eurex

Property (CFETI_EXT_PROPERTY_)	Data Type	Description
ORDER_TYPE	Uint32	The order type property defines the exchange order type of the orders. The value of this property shall have one of the following enumerated values (each defined constant shall have

		a prefix CFETI_EXT_PROPERTY_): - MARKETORDER - LIMITORDER - STOPORDER
ORDER_TIME_TYPE		The order time type defines the time restriction that shall be applied to the order. The value of this property shall have one of the following enumerated values (prefix CFETI_EXT_PROPERTY_): - ORDER_GOOD_TILL_DATE - ORDER_GOOD_FOR_DAY - ORDER_GOOD_TILL_CANCEL The default order restriction that shall be applied if none is specified is that the order shall be good for the day on which it was submitted.
ORDER_RESTRICTION		The order restriction is used to place further restriction upon the order. The following order restriction enumerations are defined (prefix CFETI_EXT_PROPERTY_) - ORDER_IMMEDIATE_OR_CANCEL - ORDER_COMPLETE_VOLUME
ORDER_EXPIRY	Uint32	Expiry date to use if the order is good until a specific date. The format of the date shall be CCYYMMDD where CC = century, YY = year, MM = month, DD = day. For example, 1st June 2003 should be specified as 20030601.
ACCOUNT_CODE	String	String used by to define properties of the account under which the order is submitted to the exchange. The following account code strings are valid: A1 Customer account G1 Give-up account, clearing user not specified G2 Give-up account, clearing user is specified M1 Market maker for customer account M2 Market maker for house proprietary account P1 Orders executed on behalf of a clearing firm P2 Orders executed on behalf of a non-member employee
LABEL	String	A free-format text label that can be attached to the order and returned in subsequent messages for that order. Note that some exchanges may impose an upper bound on the length of the string that can be used. This limit is not enforced in the eSpeed implementation.
EXCHANGE_TRADE_REF	String	The exchange trade reference is the identifier given to an order placed on the exchange. It is delivered in trade confirmations for exchange orders by eSpeed in addition to the eSpeed trade identifier assigned to each trade to uniquely define the trade on eSpeed. It remains unchanged while an order remains on the exchange, no matter how many partial fills are made against it.

		If the order is modified (in price and/or volume), however, it does change, as the exchange regards this as a new order.
TRS_ORDER_ID	String	The exchange TRS Order ID is assigned to each partial fill (trade) made against an order placed on the exchange.
EXCHANGE_ACCOUNT_NAME	String	The client application can opt to specify the account name instead of the extended properties that define the account. In this case the eSpeed trading system interface shall insert any account attributes that are not specified in the order before submitting the order to the exchange. This facility is available only where the exchange interface supports the account management function.

Trades confirmed for CME populate the instrumentIdType field in the CFETI_ORDER_DESC data structure with the value CFETI_INSTRUMENT_ID_ISIN and the instrumentId field with the instrument identifier itself.

5.3   CME

Property (CFETI_EXT_PROPERTY_)	Data Type	Description
ORDER_TYPE	Uint32	The order type property defines the exchange order type of the orders. The value of this property shall have one of the following enumerated values (each defined constant shall have a prefix CFETI_EXT_PROPERTY_): ▪ MARKETORDER ▪ LIMITORDER ▪ STOPORDER ▪ STOPLIMITORDER ▪ MARKETLIMITORDER
ORDER_TIME_TYPE		The order time type defines the time restriction that shall be applied to the order. The value of this property shall have one of the following enumerated values (prefix CFETI_EXT_PROPERTY_): ▪ ORDER_GOOD_FOR_DAY ▪ ORDER_GOOD_TILL_CANCEL The default order restriction that shall be applied if none is specified is that the order shall be good for the day on which it was submitted.
ORDER_RESTRICTION		The order restriction is used to place further restriction upon the order. The following order restriction enumerations are defined (prefix CFETI_EXT_PROPERTY_) ▪ ORDER_IMMEDIATE_OR_CANCEL

ORDER_OPENPOSITION	Uint32	Whether or not the order is to open or close a position. This attribute is not delivered in trade confirmations. If the order is to open a position the field shall have a zero value or else if it is to close a position the value shall be non-zero. If the attribute is not sent in the order then open position is assumed.
MINIMUM_VOLUME	Decimal	Minimum volume attached to the order. If the value is not specified then no minimum volume is assumed. This attribute is not delivered in trade confirmations.
ACCOUNT_CODE	String	String used by to define properties of the account under which the order is submitted to the exchange. This is a free-entry field and should be used as a representation of a back office account code that can be used by the clearing house to link the order to the actual account. This is not the same as the member code which refers to the id of the member in relation to the exchange.
CTI_CODE	Uint32	This order property defines the Customer Type Indicator. The following customer type indicator enumerations are defined (prefix CFETI_EXT_PROPERTY_CTI_CODE_) ▪ OWN_ACCOUNT ▪ HOUSE_PROPRIETARY ▪ OTHER_MEMBER ▪ CUSTOMER_ACCOUNT
CME_ORDER_ORIGIN	Uint32	This order property is specific to CME and defines the order origin. The following order origin enumerations are defined (prefix CFETI_EXT_PROPERTY_CME_ORDER_ORIGIN_). ▪ CUSTOMER ▪ FIRM ▪
CUSTOMER_ACCOUNT	String	A customer account name that can be attached to the order. Note that some exchanges may impose an upper bound on the length of the string that can be used. This limit is not enforced in the eSpeed implementation.
EXCHANGE_TRADE_REF	String	The exchange trade reference is the identifier given to an order placed on the exchange. It is delivered in trade confirmations for exchange orders by eSpeed in addition to the eSpeed trade identifier assigned to each trade to uniquely define the trade on eSpeed. It remains unchanged while an order remains on the exchange, no matter how many partial fills are made against it. If the order is modified (in price and/or volume), however, it does change, as the exchange regards this as a new order.
TRS_ORDER_ID	String	The exchange TRS Order ID is assigned to each partial fill (trade) made against an order placed on the exchange.
EXCHANGE_ACCOUNT_NAME	String	The client application can opt to specify the account name instead of the extended properties that define the account. In this case the eSpeed trading system interface shall insert any account attributes that are not specified in the order before

submitting the order to the exchange. This facility is available only where the exchange interface supports the account management function.

Trades confirmed for CME populate the instrumentIdType field in the CFETI_ORDER_DESC data structure with the value CFETI_INSTRUMENT_ID_ISIN and the instrumentId field with the instrument identifier itself.

6 Market Data

The following eCBOT-specific enumerations are defined for the price type field (CFETF_PRICETYPE). In each case the display price is delivered in the market data field with no decimal places. The published price types can be used to determine price tick behavior etc.

Pricetype (CFETI_)
PRICETYPE_CBOT_32ND
PRICETYPE_CBOT_HALF_32ND
PRICETYPE_CBOT_QTR_32ND
PRICETYPE_CBOT_64TH
PRICETYPE_CBOT_HALF_64TH
PRICETYPE_CBOT_3D_HALVES
PRICETYPE_CBOT_HALVES
PRICETYPE_CBOT_FOURTHS
PRICETYPE_CBOT_MOD_FOURTHS
PRICETYPE_CBOT_EIGHTHS
PRICETYPE_CBOT_TENTHS
PRICETYPE_CBOT_TWENTIETHS
PRICETYPE_CBOT_MOD_FORTIETHS
PRICETYPE_CBOT_HUNDREDTHS
PRICETYPE_CBOT_HALF_HUNDREDTHS