eSpeed API C/C++ Application Programming Interface

DIRECT DEALING

System Version: 1.4.13

Revision Date: April 24, 2006

1 Overview

The eSpeed electronic trading platform provides a direct dealing function that is distinct from the interactive-matching based trading that has been available from its inception. Access to the direct dealing system is available through the eSpeed API. This document presents the mechanism for accessing and using the direct dealing facility through custom applications written to the eSpeed API. Full eSpeed API documentation is available in the *C/C++ Application Program Interface: 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 Direct Dealing Registration

Before an application can make use of the eSpeed Direct Dealing facility, it is first necessary to register with the service for each business where it is available. The registration request will be acknowledged as either accepted or rejected. An accepted registration will be followed by a notification that shall deliver the Direct Dealing System Attributes, and if a callout-style market is supported, a notification to deliver the list of legal entities available to the registered user.

3.1 Availability

Whether or not direct dealing is available to a trading session is reported to the application when the trading system connection request is accepted (command CFETI_CONNECTION_ACCEPTED) through the trade flags bit-mask (`tradeFlags`) in the CFETI_CONNECT_INFO data structure.

```
/**
 * CFETI_CONNECT_INFO: Trading system session properties
 */
typedef struct CFETI_CONNECT_INFO CFETI_CONNECT_INFO;
struct CFETI_CONNECT_INFO {
    CFETI_TRADE_SESS_ID sessionId; /**< Trading system session id */
    CFETI_TRADING_SYS_DESC tradingSystem; /**< Trading system info */
    CFETI_TRADE_CONTROL_FLAGS tradeFlags; /**< Properties bit-mask */
};
```

If the bit CFETI_DIRECT_DEALING_ENABLED is set then direct dealing functionality is available. If it is not set then it is not available at the time of authentication. An application may receive, subsequent to initial authentication, a notification of a modified connection through the delivery of the command CFETI_CONNECTION_MODIFIED to the application trading system connection callback. This command is used to report changes to the trade control flags. This may include that the availability of Direct Dealing has changed.

3.2 Registration

Registration for Direct Dealing is required before the application can make any Direct Dealing requests or receive notifications of Direct Dealing requests or quotes. This is achieved through the eSpeed API *CFETIPostMessage* interface using the command CFETC_DD_REGISTER.

```
    CFETI_RC CFETIPostMessage(
            const CFETI_SESSION_ID sessId,
            CFETI_TRADE_SESS_ID trdSysSessId,
            CFETI_CMD cmd,
            CFETI_CMDDATA cmdData,
            CFETI_CMDPREF cmdPreferences );
```

`sessId`	Valid session identifier from previously successful login.
`trdSysSessId`	Trading system session identifier returned on successful connection.
`cmd`	Command (request) being submitted. To register for Direct Dealing this shall be CFETC_DD_REGISTER. (The CFETI commands that can be posted using CFETIPostMessage are

	defined in cfeti_consts.h. Each has the prefix CFETC).
`cmdData`	No command data is required for Direct Dealing registration. The application should supply 0 (zero).
`cmdPreferences`	No command preferences are available for Direct Dealing registration. The application should supply 0 (zero).

A successful return code, `CFETI_SUCCESS`, indicates that the registration request has been successfully processed. Otherwise, the reason for the failure is indicated in the return code. Return codes that may be delivered to the application include:

`CFETI_SUCCESS`	The request was processed successfully. Whether or not the posted market was accepted or rejected by the trading system is returned in the `tradingSysCallback` specified when the original connection request was made.
`CFETI_INVALID_ARG`	Invalid argument supplied to `CFETIPostMessage`.
`CFETI_NO_SESSION`	An attempt is made to post an eSpeed API request without having previously established a session with the eSpeed Session Manager.
`CFETI_NO_SUCH_LOGIN`	The login identified by the supplied session identifier cannot be found.
`CFETI_NO_SUCH_CONNECTION`	The connection specified by the supplied trading system session identifier cannot be found.

Responses to a Direct Dealing registration request are as follows:

Command Response	Command Details
Direct Dealing registration accepted	Registration was successful. No command specific data is delivered to the application for a successful registration with the Direct Dealing facility.
Direct Dealing registration rejected	The registration request was rejected. Possible reasons include: • Action prohibited (e.g. service not available)

System responses related to the Direct Dealing registration request are returned in the callback routine provided in the initial `CFETIConnect` call. The callback routine is invoked as follows:

```
void tradeSysCallback(
            CFETI_CMD cmd,
            CFETI_CMD_STATUS cmdStatus,
            CFETI_CMDDATA cmdData,
            CFETI_UD userData );
```

Commands that may be issued by the eSpeed API to the application as a result of the posted market request are as follows:

Command (CFETI_DD_)	Command Status	Command Data Type
`REGISTER_ACCEPTED`	`CFETI_SUCCESS`	N/A
`REGISTER_REJECTED`	`CFETI_ACTION_PROHIBITED`	N/A

3.3 Set System Attributes

Subsequent to successful registration for Direct Dealing the trading system connection callback for the registered trading session (provided in the initial call to CFETIConnect) shall be invoked to deliver the direct dealing system attributes to the application. The callback routine is invoked as follows:

```
    void tradeSysCallback(
                CFETI_CMD cmd,
                CFETI_CMD_STATUS cmdStatus,
                CFETI_CMDDATA cmdData,
                CFETI_UD userData );
```

Command (CFETI_DD_)	Command Status	Command Data Type
SET_SYSTEM_ATTRIBUTES	CFETI_SUCCESS	Structure CFETI_DD_SERVER_ATTRIBUTES_DESC containing an unsigned integer bit-mask describing Direct Dealing facilities available to this trading session.

The definition of the CFETI_DD_SERVER_ATTRIBUTES_DESC data structure is as follows:

```
struct CFETI_DD_DETAILS_DESC {
    unsigned int maxTimeout;     /**< Max timeout (secs) */
    unsigned int minTimeout;     /**< Min timeout in (secs) */
    unsigned int timeoutIncrement; /**< Timeout increment (secs) */
};

typedef struct CFETI_DD_SERVER_ATTRIBUTES_DESC
CFETI_DD_SERVER_ATTRIBUTES_DESC;
struct CFETI_DD_SERVER_ATTRIBUTES_DESC {
    unsigned int uiServerFeatures; /**< Server attributes bit-mask */
    double dBOLSValidSpread; /**< Maximum valid spread for BOLS */
    double dMinimumSize; /**< Minimum size for requests and quotes */
    unsigned int minAnonymousCalloutLE; /**< Minimum number of Legal
Entities required in an anonymous callout */
    CFETI_DD_DETAILS_DESC rfqDetails;
    CFETI_DD_DETAILS_DESC quoteDetails;
};
```

The system attributes bit-mask is created from the values listed below. In each case the default value that should be assumed by the application if the server attributes bit-mask value is zero is given. The BOLS valid spread value defines the maximum spread applicable to BOLS requests and quotes.

Constant (CFETI_DD_)	Description
ANONYMOUS_CALLOUT_ENABLED	Direct Dealing system supports anonymous callout functionality. (The default if the attribute bit-mask is zero is no).
NAME_GIVEUP_CALLOUT_ENABLED	Direct Dealing system supports name give-up callout functionality. (The default if the attribute bit-mask is zero is no).
2WAYDD_ENABLED	Direct Dealing system supports two-way Direct Dealing requests. (The default if the attribute bit-mask is zero is no).
1WAYDD_ENABLED	Direct Dealing system supports one-way Direct Dealing requests. (The default if the attribute bit-mask is zero is yes).
BOLS_ENABLED	Direct Dealing system supports BOLS Direct Dealing (The

	default if the attribute bit-mask is zero is no).
D2D_ENABLED	Direct Dealing system supports Dealer-to-Dealer Direct Dealing (The default if the attribute bit-mask is zero is no). This facility is mutually exclusive with BOLS direct dealing.
ALL_OR_NONE_ENABLED	Direct Dealing system is all-or-none enabled (i.e. minimum allowable quantity is not supported, default if the attribute bit-mask is zero is no).
ALLOW_RETAIN_ON_EXECUTE	Direct Dealing system allows a Direct Dealing request to be submitted with an option that it be retained on execution. (The default if the attribute bit-mask is zero is no).
ENFORCE_RETAIN_ON_EXECUTE	Direct Dealing system requires all Direct Dealing requests to be submitted with the option that it be retained on execution. (The default if the attribute bit-mask is zero is no). If this flag is set the eSpeed API shall automatically introduce this flag to all submitted Direct Dealing requests.
DISABLE_QUOTE_TIMEOUT	Direct Dealing system supports quote timeout by default. All quotes remain valid for the lifetime of the Direct Dealing or until the timeout expires. (The default if the attribute bit-mask is zero is no).
INDICATION_OF_INTEREST_ENABLED	Direct dealing system supports direct dealing requests that are an indication of interest only. Quotes will be accepted against such direct dealing requests, but cannot be traded.

3.4 Legal Entity List for Callout

If the direct dealing system attributes include that the system is enabled either for named callout or anonymous callout, the list of legal entities that shall be available to the registered trading session for callout shall be delivered to the trading system connection callback for the registered trading session subsequent to registration and confirmation of the server attributes. The callback routine is invoked as follows:

```
void tradeSysCallback(
            CFETI_CMD cmd,
            CFETI_CMD_STATUS cmdStatus,
            CFETI_CMDDATA cmdData,
            CFETI_UD userData );
```

Command (CFETI_DD_)	Command Status	Command Data Type
SET_LEGAL_ENTITY_LIST	CFETI_SUCCESS	Pointer to data structure of type CFETI_LEGAL_ENTITY_LIST_DESC containing an array of legal entity definitions.

The definition of the CFETI_LEGAL_ENTITY_LIST_DESC data structure is as follows:

```
typedef struct CFETI_LEGAL_ENTITY_DESC CFETI_LEGAL_ENTITY_DESC;
struct CFETI_LEGAL_ENTITY_DESC {
    unsigned int uiLegalEntityId;
    const char* szLegalEntityName;
    const char* szLEShortName;
```

```
};

typedef struct CFETI_LEGAL_ENTITY_LIST_DESC
CFETI_LEGAL_ENTITY_LIST_DESC;
struct CFETI_LEGAL_ENTITY_LIST_DESC {
    unsigned int uiNumEntities;
    CFETI_LEGAL_ENTITY_DESC* pEntity;
};
```

Type	Element	Description
unsigned int	uiLegalEntityId	Legal entity id
const char*	szLegalEntityName	Legal entity name
const char*	szLEShortName	Legal entity nickname
unsigned int	uiNumEntities	Number of legal entities
CFETI_LEGAL_ENTITY_DESC	pEntity	Legal entity array

When submitting direct dealing callout requests to a direct dealing system that supports anonymous or name-give-up callout it shall be necessary for the customer application to specify the list of legal entities to which the request shall be issued.

4 Direct Dealing Requests

4.1 Submitting a Request

Direct dealing requests are submitted through the eSpeed API *CFETIPostMessage* interface. Basic validation is performed on the request that may result in it being rejected by the eSpeed API. Once successfully submitted to the Direct Dealing system further validation will be carried out with the result that the request will be acknowledged as either accepted or rejected. If the request is rejected an indication of the reason will be delivered in the response.

```
CFETI_RC CFETIPostMessage(
        const CFETI_SESSION_ID sessId,
        CFETI_TRADE_SESS_ID trdSysSessId,
        CFETI_CMD cmd,
        CFETI_CMDDATA cmdData,
        CFETI_CMDPREF cmdPreferences );
```

`sessId`	Valid session identifier from previously successful login.
`trdSysSessId`	Trading system session identifier returned on successful connection.
`cmd`	Command (request) being submitted. To post a direct dealing request this shall be CFETC_DD_REQUEST_SUBMIT. To submit a subsequent request to modify a direct deal request the command shall be CFETC_DD_REQUEST_MODIFY and to submit a request to cancel a direct deal request the command shall be CFETC_DD_REQUEST_CANCEL. (The CFETI commands that can be posted using CFETIPostMessage are defined in cfeti_consts.h. Each has the prefix CFETC).
`cmdData`	Command data that contains the details of the request. The CFETIPostMessage interface is used for a number of different eSpeed API operations and the command data varies accordingly. In the case of direct dealing requests a pointer to a CFETI_DD_REQUEST_DESC data structure is expected as the command data.
`cmdPreferences`	No command preferences are available for direct dealing requests. The application should supply 0 (zero).

A successful return code, `CFETI_SUCCESS`, indicates that the direct dealing request has been successfully processed. Otherwise, the reason for the failure is indicated in the return code. Return codes that may be delivered to the application include:

`CFETI_SUCCESS`	The request was processed successfully. Whether or not the posted market was accepted or rejected by the trading system is returned in the `tradingSysCallback` specified when the original connection request was made.
`CFETI_INVALID_ARG`	Invalid argument supplied to `CFETIPostMessage`.
`CFETI_NO_SESSION`	An attempt is made to post an eSpeed API request without having previously established a session with the eSpeed Session Manager.
`CFETI_NO_SUCH_LOGIN`	The login identified by the supplied session identifier cannot be found.

`CFETI_NO_SUCH_CONNECTION`	The connection specified by the supplied trading system session identifier cannot be found.

The `CFETI_DD_REQUEST_DESC data structure` supplied to `CFETIPostMessage` for submitting, modifying or canceling direct dealing requests is defined as follows:

```
typedef unsigned char CFETI_ACTION_TYPE;
struct CFETI_DD_REQUEST_DESC {
    CFETI_ID         id;
    CFETI_ID         requestId;
    CFETI_INSTRUMENT tradeInstrument;
    CFETI_PREF       preferences;
    CFETI_SIZE       size;
    CFETI_SIZE       minSize;
    CFETI_PRICE      askPrice;
    CFETI_PRICE      bidPrice;
    CFETI_PRICE      askReservePrice;
    CFETI_PRICE      bidReservePrice;
    unsigned int     minPricesCount;
    CFETI_TIME       timeLimit;
    CFETI_TIME       timeRemaining;
    CFETI_ACTION_TYPE action;
    unsigned int     uiOwner;
    unsigned int     uiOwnerType;
    CFETI_TEXTNOTIFY pMessage;
    CFETI_PRICE      bolsSpread;
    const char*      szClassification;
    unsigned int     uiNumCounterparties;
    unsigned int*    pCounterpartyList;
    const char*      szFullUsername;
    const char*      szLegalEntityName;
    const char*      szLEShortName;
    unsigned int     uiCounterpartyStateBid;
    unsigned int     uiCounterpartyStateAsk;
    unsigned short   usInstrumentClass;
    const char*      szUserName;
    const char*      szRequestorId;
    const char*      szOriginatorId;
    unsigned int     orderInfoType;
    void*            orderInfo;
    time_t           creationTime;
    const char*      allocationInfo;
};
typedef struct CFETI_DD_REQUEST_DESC CFETI_DD_REQUEST_DESC;
typedef CFETI_DD_REQUEST_DESC* CFETI_DD_REQUEST;
```

It is strongly recommended that the application should initialize the entire direct dealing request structure to zero before filling in any of the fields. For example:

```
    CFETI_DD_REQUEST_DESC request;
    memset((char*)&request, 0, sizeof(request));
```

A description of each element of the data structure is given in the table below. Unless noted otherwise each element shall be populated in all direct dealing requests and included in responses to those requests and also in notifications of the requests to users of the system.

Type	Element	Description
CFETI_ID	id	ID of the Direct Dealing request. This shall be zero if the command is CFETC_DD_REQUEST_SUBMIT or the value assigned by the Direct Dealing system if the command is CFETC_DD_REQUEST_MODIFY or CFETC_DD_REQUEST_DELETE.
CFETI_ID	requestId	This shall be non-zero if the command is CFETC_DD_REQUEST_SUBMIT or zero if the command is CFETC_DD_REQUEST_MODIFY or CFETC_DD_REQUEST_DELETE.
CFETI_INSTRUMENT	tradeInstrument	ESpeed instrument identifier for the instrument for which a quote is requested
CFETI_PREF	preferences	Direct dealing specific options for this request
CFETI_SIZE	size	Absolute size that the user is requesting
CFETI_SIZE	minSize	Minimum size that the user is requesting
CFETI_PRICE	askPrice	Price that the requestor is quoting (Reserved for future use)
CFETI_PRICE	bidPrice	Price that the requestor is quoting (Reserved for future use)
CFETI_PRICE	askReservePrice	Reserve price (Reserved for future use)
CFETI_PRICE	bidReservePrice	Reserve price (Reserved for future use)
unsigned int	minPricesCount	Minimum number of prices that the user requires. (Reserved for future use).
CFETI_TIME	timeLimit	Number of seconds for which the request/quote will be valid (specified by the creator)
CFETI_TIME	timeremaining	Number of seconds for which the request/quote will remain valid (specified by the direct dealing system when notifying users of the request)
CFETI_ACTION_TYPE	action	Action (CFETI_DD_BUY, CFETI_DD_SELL, CFETI_DD_2WAY, CFETI_DD_BOLS_2WAY)
unsigned int	uiOwner	Id of the Direct Dealing request owner. Only valid in responses or notifications. Not valid in user requests.
unsigned int	uiOwnerType	Enumerated constant to indicate whether the owner of the request is the receiving user, in the same legal entity or some other counterparty. Only valid in responses or notifications. Not valid in user requests.
CFETI_TEXTNOTIFY	pMessage	Text message associated with this request. Only valid in notifications when the command is CFETI_DD_SYSTEM_MESSAGE).
CFETI_PRICE	bolsSpread	BOLS spread to use in two-way BOLS requests.

`const char*`	`szClassification`	Instrument classification string for the trade instrument. Only valid in responses or notifications. Not valid in user requests.
`unsigned int`	`uiNumCounterparties`	Number of counterparties in the legal entity list (specific to anonymous or named callout).
`unsigned int*`	`pCounterpartyList`	List of legal entity ids for this Direct Deal request (specific to anonymous or named callout)
`const char*`	`szFullUsername`	Full user name of the Direct Deal request owner. Only valid in responses or notifications. Not valid in user requests.
`const char*`	`szLegalEntityName`	Full legal entity name of the Direct Deal request owner. Only valid in responses or notifications. Not valid in user requests.
`const char*`	`szLEShortName`	Nickname of the legal entity of the Direct Deal request owner. Only valid in responses or notifications. Not valid in user requests.
`unsigned int`	`uiCounterpartyStateBid`	Enumerated counterparty state for the bid price (specific to businesses where name giveup is enabled). Only valid in responses or notifications. Not valid in user requests.
`unsigned int`	`uiCounterpartyStateAsk`	Enumerated counterparty state for the ask price (specific to businesses where name giveup is enabled). Only valid in responses or notifications. Not valid in user requests.
`unsigned short`	`usInstrumentClass`	Instrument classification value (specific to businesses where name giveup is enabled). Required in user requests for businesses where name giveup is enabled.
`const char*`	`szUserName`	Name of the user to whom the request belongs. Not required in user requests.
`const char*`	`szRequestorId`	Name of the user that submitted the request. Not valid in user requests.
`const char*`	`szOriginatorId`	Name of the user that submitted the original request. Not valid in user requests.
`unsigned int`	`orderInfoType`	Enumerated constant to indicate data structure that orderInfo points to.
`void*`	`orderInfo`	Pointer to data structure specified by orderInfoType. Not valid if orderInfoType is CFETI_ ORDERINFO_NOT_SPECIFIED.
`time_t`	`creationTime`	The time at which the request was created, expressed as a number of seconds since 00:00:00 1/1/70 GMT. If not available for the business in question the value shall be zero. The

		field is ignored in all submitted requests and is populated only in notifications and responses.
`const char*`	`allocationInfo`	Free text for allocation information that will be preserved with the request in notifications and any subsequent trade confirmation.

In a give-up enabled business, the availability of the facility to bid or ask to the receiver of the direct dealing request is indicated through the uiCounterpartyStateBid and uiCounterpartyStateAsk fields of the direct dealing request structure. Each of these is assigned one of the following enumerated values. For such businesses it is a requirement that the instrument classification value for that instrument (as provided in the market data stream) is delivered in all direct dealing requests for the instrument.

Constant	Description
CFETI_OWNER_TYPE_NONE	Owner type is not specified (e.g. instrument or business is not name-giveup enabled).
CFETI_OWNER_TYPE_THIS_USER	This user is the owner of the request / quote
CFETI_OWNER_TYPE_THIS_LEGAL_ENTITY	This user is in the same legal entity as the owner of the request / quote
CFETI_OWNER_TYPE_UNKNOWN	This user is not in the same legal entity as the owner of the request / quote.

The preferences that can be specified in the direct dealing request preferences field are as described in the table below:

Constant	Description
CFETI_DD_RETAIN_ON_EXECUTE	The direct dealing request should be retained by the direct dealing server and re-advertised after it has been filled.
CFETI_DD_IS_ANONYMOUS_CALLOUT	The direct dealing request is an anonymous callout request.
CFETI_DD_IS_NAMED_CALLOUT	The direct dealing request is a named callout request.
CFETI_DD_IS_INDICATION_OF_INTEREST	The direct dealing request is an indication of interest. Quotes will be accepted against such direct dealing requests, but cannot be traded.

The following enumerations for orderInfoType are valid.

Constant	Description
CFETI_ORDERINFO_PV01_LOCK	CFETI_PV01_LOCK_DESC
CFETI_ORDERINFO_CANTOR_REPO	CFETI_CANTOR_REPO_DESC

If the orderInfoType value is CFETI_ORDERINFO_PV01_LOCK then the orderInfo pointer will contain the address of a CFETI_PV01_LOCK_DESC data stucture. This structure is defined as follows.

```
struct CFETI_TSWAP_DESC {
    double   dTSwapRatio;        /**< Selected PV01 */
    double   dLockPrice;         /**< Selected lock price */
    double   dCurrentTSwapRatio; /**< Prevailing PV01 */
    double   dCurrentLockPrice;  /**< Prevailing lock price */
};
```

When submitting a request for a US Treasury Swap or for an Interest Rate Swap vs. Future instrument this data structure is used – but need not be populated in direct dealing requests. When responses to the direct

dealing request are issued they shall include the prevailing PV01 and Lock Price at the time that the request was first submitted shall be delivered. Quotes entered against the request shall be normalized to this PV01 and Lock Price.

If the orderInfoType value is CFETI_ORDERINFO_CANTOR_REPO then the orderInfo pointer will contain the address of a CFETI_CANTOR_REPO_DESC data structure. This structure is defined as follows:

```
struct CFETI_CANTOR_REPO_DESC {
    unsigned int startDate;  /**< Repo start date (CCYYMMDD) */
    unsigned int endDate;    /**< Repo start date (CCYYMMDD) */
    unsigned int rightsOfSubstitution; /**< Rights of sub */
    const char* annotation;  /**< Annotation text */
    unsigned int attributes; /**< Attributes bit-mask */
};
```

The start date field is the start date of the repo in the format CCYYMMDD. Similarly the end date is the end date of the repo again in the format CCYYMMDD. A special value of zero shall be valid for the end-date to indicate that it is open. The rights of substitution field allows the requestor to specify the number of collateral substitutions that are required. The direct dealing server shall give greater merit to quotes that are closest to the requested number of substitutions when ranking quotes received. The annotation field allows the requestor to attach text to the direct dealing request (or quote). Modifications to requests and quotes will update the text if different text is specified. If no text is supplied then the existing text will retained by the direct dealing server. The text will be included in notifications to other users of the requests and quotes.

The following values are defined for the attributes bit-mask. Requests that either specify neither money-fill nor face-value shall be rejected, as shall requests that specify both.

Constant	Description
CFETI_DD_REPO_MONEY_FILL	Request is a money-fill request.
CFETI_DD_REPO_FACE_VALUE	Request is a face-value request.

The user that submitted the direct dealing request shall receive a response to indicate whether or not the request was accepted or rejected. The table below indicates the possible responses according to the command that was specified in the request.

User command	Response command
CFETC_DD_REQUEST_SUBMIT	CFETI_DD_REQUEST_SUBMIT_ACCEPTED
CFETC_DD_REQUEST_SUBMIT	CFETI_DD_REQUEST_SUBMIT_REJECTED
CFETC_DD_REQUEST_CANCEL	CFETI_DD_REQUEST_CANCEL_ACCEPTED
CFETC_DD_REQUEST_CANCEL	CFETI_DD_REQUEST_CANCEL_REJECTED
CFETC_DD_REQUEST_MODIFY	CFETI_DD_REQUEST_MODIFY_ACCEPTED
CFETC_DD_REQUEST_MODIFY	CFETI_DD_REQUEST_MODIFY_REJECTED

In each case the response shall be delivered to the trading system connection callback for the connection under which the request was made. If the response is to reject the direct dealing request, a command status field shall indicate the reason for the rejection. A full list of direct dealing specific command status codes is given later in this document. A message to cancel a direct dealing request will be rejected if it is a dealer-to-dealer request and there is already a qualifying quote against the direct dealing request. In this case the command status code shall be CFETI_DD_REQUEST_SPREAD_MET.

4.2 Notifications

Once a direct dealing request has been accepted by the system it will be advertised to other users of the system. In a call-out market (anonymous or name give-up) the users that receive notification of the request shall be those to whom the request is advertised. For other market places the direct dealing request shall be advertised to all users that will have the opportunity to provide a quote. If the Direct Dealing request is modified as a result of a modify request by the requestor, or if the direct dealing request is subsequently cancelled, either because its time of validity has expired, or as the result of a successful cancel request by the requestor, the same users shall be notified. Similarly, if the direct dealing request results in a trade, an indication of execution shall be sent to the requestor and the supplier(s) of the quotes executed.

The list of direct dealing request notifications and a description of their interpretation is given in the table below. For each notification the command data shall be a pointer to a populated direct dealing request data structure containing a full description of the direct dealing request.

System Command (CFETI_DD_)	Description
REQUEST_CREATED	A new direct dealing request (advertised to the receiver of the notification and the originator of the request)
REQUEST_MODIFIED	The direct dealing request described in the command data has been modified – the new details of the request or indicated in the notification. If a dealer-to-dealer direct dealing request is in receipt of a qualifying spread this notification shall be used and the command status code shall be CFETI_DD_REQUEST_SUBMIT. If a qualifying spread is cancelled by the quote provider before it can be accepted by the requestor a further request modified notification shall be issued, this time with the command status code CFETI_SUCCESS. A request modified notification is also issued if a quote is received with a timeout that is greater than the amount of time remaining in the request to extend the timeout of the request by the difference.
REQUEST_CANCELLED	The direct dealing request described in the command data has been cancelled. The command status structure delivered to the trading system connection callback may indicate the reason for the cancellation.
REQUEST_EXECUTED	This notification is sent only to the original requestor and to providers of quotes that the requestor has accepted. Detail on the mechanism for accepting quotes is given later in this document.
SYSTEM_MESSAGE	The system message notification is sent to deliver a text message to the user for the Direct Deal request in question.
REQUEST_REFRESH	This notification is sent to refresh the user's image of the direct dealing request.

When the response command is CFETI_DD_SYSTEM_MESSAGE the Direct Dealing request structure shall include a pointer to a text notification containing a text message delivered to the receiver. This structure is defined as follows:

```
struct CFETI_TEXTNOTIFY_DESC {
    CFETI_TRADING_SYSTEM        tsId;
    unsigned int                severity;
    time_t                      sendTime;
    char**                      text;
    int                         textLines;
    char*                       sender;
};
typedef struct CFETI_TEXTNOTIFY_DESC CFETI_TEXTNOTIFY_DESC;
typedef struct CFETI_TEXTNOTIFY_DESC* CFETI_TEXTNOTIFY;
```

A description of each element of the data structure is given in the table below

Type	Element	Description
`CFETI_TRADING_SYSTEM`	`tsId`	eSpeed trading system identifier

`unsigned int`	`severity`	Message severity. This shall have one of the following values:
		`CFETI_TEXT_SEVERITY_UNKNOWN`
		`CFETI_TEXT_SEVERITY_LOW`
		`CFETI_TEXT_SEVERITY_MEDIUM`
		`CFETI_TEXT_SEVERITY_HIGH`
		`CFETI_TEXT_SEVERITY_URGENT`

`time_t`	`sendTime`	Time that the notification was sent (seconds since 1/1/1970 00:00:00 GMT).
`char**`	`text`	Array of lines of text in the message
`int`	`textLines`	Number of lines of text in the array
`char*`	`sender`	Details of the message sender (not guaranteed to be set)

5 Direct Dealing Quotes

5.1 Submitting a Quote

Quotes sent in response to direct dealing requests are also submitted through the eSpeed API *CFETIPostMessage* interface. Basic validation is performed on the quote that may result in it being rejected by the eSpeed API. Once successfully submitted to the Direct Dealing system further validation will be carried out with the result that the quote will be acknowledged as either accepted or rejected. If the quote is rejected an indication of the reason will be delivered in the response.

```
CFETI_RC CFETIPostMessage(
        const CFETI_SESSION_ID sessId,
        CFETI_TRADE_SESS_ID trdSysSessId,
        CFETI_CMD cmd,
        CFETI_CMDDATA cmdData,
        CFETI_CMDPREF cmdPreferences );
```

`sessId`	Valid session identifier from previously successful login.
`trdSysSessId`	Trading system session identifier returned on successful connection.
`cmd`	Command (request) being submitted. To post a direct dealing quote this shall be CFETC_DD_QUOTE_SUBMIT. To submit a subsequent request to modify a direct deal request the command shall be CFETC_DD_QUOTE_MODIFY and to submit a request to cancel a direct deal request the command shall be CFETC_DD_REQUOTE_CANCEL. (The CFETI commands that can be posted using CFETIPostMessage are defined in cfeti_consts.h. Each has the prefix CFETC).
`cmdData`	Command data that contains the details of the request. The CFETIPostMessage interface is used for a number of different eSpeed API operations and the command data varies accordingly. In the case of direct dealing quotes a pointer to a CFETI_DD_QUOTE_DESC data structure is expected as the command data.
`cmdPreferences`	Command preferences are available for direct dealing BOLS and dealer-to-dealer quotes. These preferences allow the quoter to indicate preference to buy or sell in the event that the requestor has failed to initiate a trade. One or other of CFETI_DD_BOLS_QUOTER_FAVOURS_BUY or CFETI_DD_BOLS_QUOTER_FAVOURS_SELL shall be specified. The application should supply 0 (zero) for any quote.

A successful return code, `CFETI_SUCCESS`, indicates that the direct dealing quote has been successfully processed. Otherwise, the reason for the failure is indicated in the return code. Return codes that may be delivered to the application include:

`CFETI_SUCCESS`	The request was processed successfully. Whether or not the posted market was accepted or rejected by the trading system is returned in the `tradingSysCallback` specified when

	the original connection request was made.
CFETI_INVALID_ARG	Invalid argument supplied to CFETIPostMessage.
CFETI_NO_SESSION	An attempt is made to post an eSpeed API request without having previously established a session with the eSpeed Session Manager.
CFETI_NO_SUCH_LOGIN	The login identified by the supplied session identifier cannot be found.
CFETI_NO_SUCH_CONNECTION	The connection specified by the supplied trading system session identifier cannot be found.

The CFETI_DD_QUOTE_DESC data structure supplied to CFETIPostMessage for submitting, modifying or canceling direct dealing quotes is defined as follows:

```
struct CFETI_DD_QUOTE_DESC {
    CFETI_ID          id;
    CFETI_ID          requestId;
    CFETI_INSTRUMENT  tradeInstrument;
    CFETI_PREF        preferences;
    CFETI_SIZE        size;
    CFETI_PRICE       askPrice;
    CFETI_PRICE       bidPrice;
    CFETI_TIME        timeLimit;
    CFETI_TIME        timeRemaining;
    CFETI_ACTION_TYPE action;
    CFETI_ID          RFQId;
    const char*       szFullUsername;
    const char*       szLegalEntityName;
    const char*       szLEShortName;
    unsigned int      uiOwner;
    unsigned int      uiOwnerType;
    CFETI_PRICE       quoteSpread;
    unsigned int      uiCounterpartyStateBid;
    unsigned int      uiCounterpartyStateAsk;
    unsigned short    usInstrumentClass;
    const char*       szUserName;
    const char*       szRequestorId;
    const char*       szOriginatorId;
    unsigned int      orderInfoType;
    void*             orderInfo;
    time_t            creationTime;
    const char*       allocationInfo;
};
typedef struct CFETI_DD_QUOTE_DESC CFETI_DD_QUOTE_DESC;
typedef CFETI_DD_QUOTE_DESC* CFETI_DD_QUOTE;
```

It is strongly recommended that the application should initialize the entire direct dealing request structure to zero before filling in any of the fields. For example:

```
    CFETI_DD_QUOTE_DESC quote;
    memset((char*)&quote, 0, sizeof(quote
```

A description of each element of the data structure is given in the table below. Unless noted otherwise each element shall be populated in all direct dealing quotes and included in responses to those quotes and also in notifications of the quotes to users of the system.

Type	Element	Description

CFETI_ID	id	ID of the Direct Dealing quote. This shall be zero if the command is CFETC_DD_QUOTE_SUBMIT or the value assigned by the Direct Dealing system if the command is CFETC_DD_QUOTE_MODIFY or CFETC_DD_QUOTE_DELETE.
CFETI_ID	requestId	This shall be non-zero if the command is CFETC_DD_QUOTE_SUBMIT or zero if the command is CFETC_DD_QUOTE_MODIFY or CFETC_DD_QUOTE_DELETE.
CFETI_INSTRUMENT	tradeInstrument	ESpeed instrument identifier for the instrument for which a quote is provided
CFETI_PREF	preferences	Direct dealing specific options for this request
CFETI_SIZE	size	Absolute size that the user is quoting
CFETI_PRICE	askPrice	Ask price that the user is quoting
CFETI_PRICE	bidPrice	Bid price that the user is quoting
CFETI_TIME	timeLimit	Number of seconds for which the quote will be valid (specified by the creator)
CFETI_TIME	timeremaining	Number of seconds for which the quote will remain valid (specified by the direct dealing system when notifying users of the quote)
CFETI_ACTION_TYPE	action	Action (CFETI_DD_BID, CFETI_DD_ASK, CFETI_DD_2WAY_QUOTE, CFETI_DD_BOLS_2WAY_QUOTE)
const char*	szFullUsername	Full user name of the quote owner. Only valid in responses or notifications. Not valid in user requests.
const char*	szLegalEntityName	Full legal entity name of the owner. Only valid in responses or notifications. Not valid in user requests.
const char*	szLegalEntityName	Nickname of the legal entity of the quote owner. Only valid in responses or notifications. Not valid in user requests.
unsigned int	uiOwner	Id of the Direct Dealing quote owner. Only valid in responses or notifications. Not valid in user requests.
unsigned int	uiOwnerType	Enumerated constant to indicate whether the owner of the quote is the receiving user, in the same legal entity or some other counterparty. Only valid in responses or notifications. Not valid in user requests.
CFETI_PRICE	quoteSpread	Quote spread for 2 Way and BOLS quotes.
unsigned int	uiCounterpartyStateBid	Enumerated counterparty state for the bid price (specific to businesses where name giveup is enabled). Only valid in responses or notifications. Not valid in user requests.

`unsigned int`	`uiCounterpartyStateAsk`	Enumerated counterparty state for the ask price (specific to businesses where name giveup is enabled). Only valid in responses or notifications. Not valid in user requests.
`unsigned short`	`usInstrumentClass`	Instrument classification value (specific to businesses where name giveup is enabled). Required in user requests for businesses where name giveup is enabled.
`const char*`	`szUserName`	Name of the user to whom the quote belongs. Not required in user quotes.
`const char*`	`szRequestorId`	Name of the user that submitted the quote. Not valid in user quotes.
`const char*`	`szOriginatorId`	Name of the user that submitted the original quote. Not valid in user quotes.
`unsigned int`	`orderInfoType`	Enumerated constant to indicate data structure that orderInfo points to.
`void*`	`orderInfo`	Pointer to data structure specified by orderInfoType. Not valid if orderInfoType is CFETI_ ORDERINFO_NOT_SPECIFIED.
`time_t`	`creationTime`	The time at which the quote was created, expressed as a number of seconds since 00:00:00 1/1/70 GMT. If not available for the business in question the value shall be zero. The field is ignored in all submitted quotes and is populated only in notifications and responses.
`const char*`	`allocationInfo`	Free text for allocation information that will be preserved with the request in notifications and any subsequent trade confirmation.

The preferences that can be specified in the direct dealing quote preferences field are as described in the table below:

Constant	Description
CFETI_DD_QUOTER_FAVORS_BUY	Provider of the two-way or BOLS quote expresses a preference to buy
CFETI_DD_QUOTER_FAVORS_SELL	Provider of the two-way or BOLS quote expresses a preference to sell
CFETI_DD_QUOTE_REVERSE_INQUIRY	Quote is a reverse inquiry
CFETI_DD_QUOTE_USE_SURPLUS	Quote can be advertised as a reverse inquiry if there is no request to match

The following enumerations for orderInfoType are valid.

Constant	Description
CFETI_ORDERINFO_PV01_LOCK	CFETI_PV01_LOCK_DESC
CFETI_ORDERINFO_CANTOR_REPO	CFETI_CANTOR_REPO_DESC

If the orderInfoType value is CFETI_ORDERINFO_PV01_LOCK then the orderInfo pointer will contain the address of a CFETI_PV01_LOCK_DESC data stucture. This structure is defined in section 4.1 of this document.

When submitting a quote for a US Treasury Swap or for an Interest Rate Swap vs. Future instrument this data structure is required. The quote provider shall populate the *dTSwapRatio* and *dLockPrice* fields. The direct dealing server shall include these in the responses for this quote along with the PV01 and Lock Price prevailing when the original request was submitted. The quote provider's quote is recalculated according the prevailing PV01 and Lock Price.

If the orderInfoType value is CFETI_ORDERINFO_CANTOR_REPO then the orderInfo pointer will contain the address of a CFETI_CANTOR_REPO_DESC data structure. This structure is defined in section 4.1 of this document.

When submitting a quote for a repo this dat structure is required. The quote provider shall populate the data structure with the repo specific details of the quote. The direct dealing server shall include these in the responses for this quote and in notifications to other users.

The user that submitted the direct dealing quote shall receive a response to indicate whether or not the quote was accepted or rejected. The table below indicates the possible responses according to the command that was specified in the quote.

User command	Response command
CFETC_DD_QUOTE_SUBMIT	CFETI_DD_QUOTE_SUBMIT_ACCEPTED
CFETC_DD_QUOTE_SUBMIT	CFETI_DD_QUOTE_SUBMIT_REJECTED
CFETC_DD_QUOTE_CANCEL	CFETI_DD_QUOTE_CANCEL_ACCEPTED
CFETC_DD_QUOTE_CANCEL	CFETI_DD_QUOTE_CANCEL_REJECTED
CFETC_DD_QUOTE_MODIFY	CFETI_DD_QUOTE_MODIFY_ACCEPTED
CFETC_DD_QUOTE_MODIFY	CFETI_DD_QUOTE_MODIFY_REJECTED

In each case the response shall be delivered to the trading system connection callback for the connection under which the quote request was made. If the response is to reject the direct dealing quote request, a command status field shall indicate the reason for the rejection. A full list of direct dealing specific command status codes is given later in this document.

5.2 Notifications

Once a direct dealing quote has been accepted by the system it will be advertised to the originator of the direct dealing request and the provider of the quote.

If the Direct Dealing quote is modified as a result of a modify request, or if the direct dealing quote is subsequently cancelled, either because its time of validity has expired, or as the result of a successful cancel request, the same users shall be notified. Similarly, if the direct dealing quote results in a trade, an indication of execution shall be sent to the requestor and the supplier(s) of the quotes executed.

The list of direct dealing request notifications and a description of their interpretation is given in the table below. For each notification the command data shall be a pointer to a populated direct dealing request data structure containing a full description of the direct dealing request.

System Command (CFETI_DD_)	Description
QUOTE_CREATED	A new direct dealing quote (advertised to the provider of the quote and the originator of the direct dealing request)
QUOTE_MODIFIED	The direct dealing quote described in the command data has been

	modified – the new details of the quote or indicated in the notification.
QUOTE_CANCELLED	The direct dealing quote described in the command data has been cancelled. The command status structure delivered to the trading system connection callback may indicate the reason for the calculation.
QUOTE_EXECUTED	This notification is sent only to the original requestor and to the quote providers of quotes that the requestor has accepted. Detail on the mechanism for accepting quotes is given later in this document.
QUOTE_REFRESH	This notification is sent to refresh the user's image of the direct dealing quote.
QUOTE_ACCEPTED	This notification is received when the direct dealing requestor accepts a quote.
QUOTE_REJECTED	This notification is received when the direct dealing requestor rejects a quote.

An additional notification is delivered to the submitter of a Direct Deal request. This is to provide ranking details to the application to indicate the relative rank of quotes. Bid or offer quotes are ranked according to best bid/ask price order and if two quotes have the same price they are ranked in time order. Two-way quotes are ranked in time order. BOLS quotes are ranked according to the spread in the quote and if quotes with the same spread are ranked in time order.

System Command (CFETI_DD_)	Description
UPDATE_QUOTE_RANK	The ranking of the quotes received against the request has been modified according to the rules defined by the eSpeed Direct Deal system. The trading system connection callback is invoked with this command the command data (cmdData) shall be a pointer to a CFETI_DD_RANK_DESC data structure.

```
typedef struct CFETI_DD_QUOTE_RANK_DESC CFETI_DD_QUOTE_RANK_DESC;
struct CFETI_DD_QUOTE_RANK_DESC
{
    unsigned int uiQuoteId;
    unsigned int uiSortRank;
    unsigned int uiBidRank;
    unsigned int uiAskRank;
};

typedef struct CFETI_DD_RANK_DESC CFETI_DD_RANK_DESC;
struct CFETI_DD_RANK_DESC
{
    const char*  szInstName;
    unsigned int uiRFQId;
    unsigned int uiBOLSMatchedAsk;
    unsigned int uiBOLSMatchedBid;
    unsigned int uiCountdownTime;
    unsigned int uiNumQuoteRanks;
    CFETI_DD_QUOTE_RANK_DESC* pQuoteRanks;
};
```

A description of each element of these data structures is given in the table below.

Type	Element	Description
`unsigned int`	`id`	Id of the Direct Deal request
`const char*`	`szInstName`	Instrument identified in the Direct

		Deal request
`unsigned int`	`uiBolsMatchedAsk`	Id of the quote that matched the ask side of the BOLS spread. This shall be zero if the Direct Deal request is not BOLS.
`unsigned int`	`uiBolsMatchedBid`	Id of the quote that matched the bid side of the BOLS spread. This shall be zero if the Direct Deal request is not BOLS.
`unsigned int`	`uiCountdownTime`	Countdown time to buy or sell if the Direct Deal request is BOLS, zero otherwise.
`unsigned int`	`uiNumQuoteRanks`	Number of quote rank data structures
`CFETI_DD_QUOTE_RANK_DESC*`	`pQuoteRanks`	Pointer to array of quote rank data structures
`unsigned int`	`uiQuoteId`	The id of the ranked quote
`unsigned int`	`uiSortRank`	The display rank for the quote (Highest rank is 1)
`unsigned int`	`uiBidRank`	The relative rank of the bid for this quote against other bids (Highest rank is 1)
`unsigned int`	`uiAskRank`	The relative rank of the offer for this quote against other offers (Highest rank is 1)

6 Negotiation and Trading

Once a direct dealing request has been issued and one or more quotes received by the requestor, it is incumbent upon the originator of the direct dealing request to decide whether or not to accept one or more of the quotes provided. The facility is provided for the requestor to indicate whether to accept or reject the quotes for each quote available.

The user that submitted the direct dealing request shall receive a response to indicate whether or not the accept or reject of the quote was itself accepted or rejected. The user that submitted the quote shall receive a notification that a quote has been accepted or rejected. The table below indicates the possible responses according to the command that was specified in the request.

User command	Response command & Description
CFETC_DD_QUOTE_ACCEPT	CFETI_DD_ACCEPT_QUOTE_ACCEPTED Request to accept the quote was successful. The provider of the quote shall receive a CFETI_DD_QUOTE_ACCEPTED command. This will result in the notification of execution and delivery of a trade confirmation (CFETI_TRADE_CONFIRM) to the originating requestor, the provider of the quote and their respective straight-through-processing trade feed implementations. If a quote is accepted the requestor and provider of the quote shall both receive a CFETI_DD_QUOTE_EXECUTED command to the trading system connection callback with the details of the quote supplied as the command data. If a direct deal has resulted in a trade the requestor and the provide of the quote shall both also receive a CFETI_DD_REQUEST_TRADED command to the trading system connection callback with the details of the request supplied as the command data. If the direct dealing system attributes are that retain on execution feature is not available or if that option was not specified in the request, the direct dealing request is no longer available after execution. The requestor and all providers of quotes shall then receive a CFETI_DD_REQUEST_CANCELLED command to the trading system callback. The requestor and the providers of other quotes shall also receive CFETI_DD_QUOTE_CANCELLED notifications for each quote that was not traded. If the direct dealing system attributes are that retain on execution feature is available and that option was specified in the request, the direct dealing request remains valid after the execution. In this case the direct dealing request is cancelled only as a result of user request or when its time of validity has expired. When a trade confirm is generated that is a result of a trade negotiated using direct dealing the `tradeProperties` bit-mask field in the CFETI_ORDER_DESC data structure delivered to the application and trade feed to describe the trade shall include the bit CFETI_TRADE_DIRECT_DEAL.
CFETC_DD_QUOTE_ACCEPT	CFETI_DD_ACCEPT_QUOTE_REJECTED

	Request to accept the quote was not successful. The command status code may indicate the reason why the request was rejected.
CFETC_DD_QUOTE_REJECT	CFETI_DD_REJECT_QUOTE_ACCEPTED Request to reject the quote was successful. The provider of the quote shall receive a CFETI_DD_QUOTE_REJECTED command. The provider of the quote and the originator of the direct dealing request shall also receive a CFETI_DD_QUOTE_CANCELLED command to the trading system connection callback to indicate the cancellation of the quote.
CFETC_DD_QUOTE_REJECT	CFETI_DD_REJECT_QUOTE_REJECTED Request to reject the quote was not successful. The command status code may indicate the reason why the request was rejected.

In each case the responses and subsequent notifications shall be delivered to the trading system connection callback for the connection under which the request was made. If the response is to reject the request to accept or reject the quote, a command status field shall indicate the reason for the rejection. A full list of direct dealing specific command status codes is given later in this document.

7 Command Status Codes

When direct dealing requests are rejected, a pointer to a command status data structure (CFETI_CMD_STATUS_DESC) is delivered to the eSpeed API trading system connection callback. This data structure includes an unsigned integer status code and an optional description of the nature of the error. Listed below are the command status codes that are specific to direct dealing.

Status Code (CFETI_)	Description
DD_REQUEST_EXPIRED	The Direct Deal Request has expired - time limit reached
DD_QUOTE_EXPIRED	The Direct Deal Quote has expired - time limit reached
DD_QUOTE_CANCELLED_BY_TRADE	Direct Deal Quote is cancelled because Direct Deal Request has been traded
DD_REQUEST_INVALID_ID	Invalid Direct Dealing Request id submitted
DD_NOT_OWNER	Not owner of this Direct Deal Request/Quote
DD_MAX_REACHED	Maximum number of Direct Deal Requests/Quotes reached
DD_OUTSTANDING_REQUEST	Already have an outstanding Direct Deal Request/Quote
DD_INVALID_PARAMETERS	Values in Direct Deal Request/Quote specification are invalid
DD_INVALID_COMMAND	Invalid Direct Deal Request/Quote command
DD_INVALID_COUNTERPARTY	Cannot quote to that counterparty
DD_INVALID_INSTRUMENT	Unknown instrument
DD_USER_NOT_REGISTERED	User that issued Direct Deal Request/Quote is not registered
DD_UNABLE_TO_COMPLETE	Unable to commit Direct Deal operation
DD_LOCKED_DOWN	No modifications allowed during lock down period
DD_TRADE_NOT_APPROVED	The GiveUp Matrix does not approve a Direct Deal trade
DD_REQUEST_SPREAD_MET	The spread specified by the requestor in a BOLS or Dealer-to-Dealer request has been met.

8 Market Data

eSpeed API applications that subscribe to instrument market data receive that data as an array of fields. Each element of this array is comprised of a field identifier, a field data type and a field value. The field identifiers are defined in the eSpeed API include file `cfeti_fields.h`. The field data types are defined in the eSpeed API include file `cfeti_consts.h`. Listed below are the eSpeed API market data fields that are specific to direct dealing.

Field Identifier (CFETF_)	Data Type	Description
DD_INST_PROPERTY	uint32	Bit-mask of direct dealing specific instrument properties. Possible values are defined later in this section.
DD_REQUEST_DURATION	uint32	Default number of seconds that a direct dealing request should be valid for. If the field is not present or if the value is zero, the application shall assume a default value of thirty seconds.
DD_LOCKOUT_DURATION	uint32	Number of seconds before a direct dealing request expires during which no more quotes will be accepted (default if not present is zero).
DD_SYSTEM_ATTRIBUTES	uint32	Bit-mask of direct dealing system attributes for the instrument. If this field is present it overrides the system value provided when the registration with the direct dealing system was accepted.
DD_MIN_QUANTITY	decimal	Default minimum quantity for direct dealing that responders to a request can quote. (Actual minimum is this value multiplied by value in the field CFETF_SIZE_MULTIPLIER for the instrument).
DD_QUOTE_SPREAD	decimal	Tolerance within which two-way and BOLS quotes must fall in order to be valid. Specified as a spread from the bid price. If this value is zero then no maximum sp[read is imposed.

The following enumerations for the field CFETF_DD_INST_PROPERTY are defined.

Constant	Description
CFETI_DD_INSTRUMENT_ENABLED	Instrument is enabled for direct dealing
CFETI_DD_REQUESTER_MUST_QUOTE	Requestor must also provide a quote for the market
CFETI_DD_QUOTE_MODIFICATION_DISABLED	Modification of quotes is disabled for the instrument – only create and delete operations are supported.