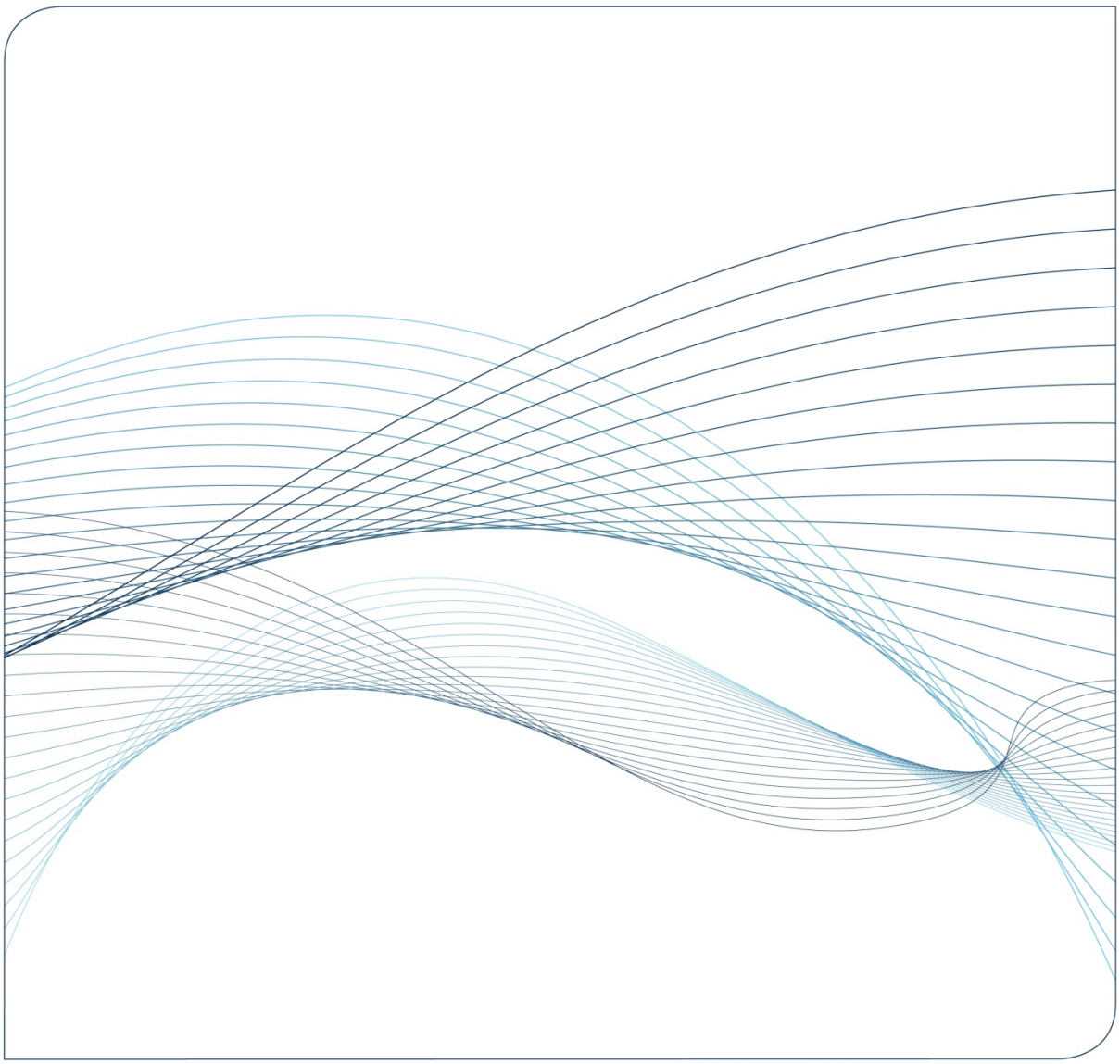
IRESS MARKET FEED

TECHNICAL OVERVIEW

Version 1.2



CONTENTS

DOCUMENT VERSION HISTORY

Protocol Version	Change Date	Changes
1.0	Jan 2011	Initial version
1.1	Nov 2011	▪ The error code in Exchange Information Message is explained. ▪ Corrections on the sizes of fields including Security Type, Bid Data Source, Ask Data Source in Quote Full Message are made. ▪ Additional explanations on individual messages including Quote Full Message, Depth Messages and Depth Consolidated Order Message are added. ▪ The possible values of Action Flag field in Trade Message is explained. ▪ Request Flag in Login Request Packet is changed to be reserved.
1.2	Dec 2011	▪ Architecture section renamed to Communication Sequence. The basic communication sequence is explained in this new section. ▪ Some paragraphs about handshaking moved from overview to Communication Sequence ▪ Recovery Sequence Messages are introduced. ▪ The Request Flag field in Login Request Packet is now used for the client to notify the server the version of Iress Market Specification the client supports. ▪ The Server Version field in Server Information Packet is now using a special format to represent the highest Iress Market Feed Specification version that the server supports.
1.2	Mar 2012	▪ Appendixes listing flags and types are added. ▪ Corrected the misprint feed port numbers.
1.2	Jul 2012	▪ Corrected the description of recovery sequence messages.

OVERVIEW

IRESS Market Feed is a low latency, normalised and consolidated market data feed that is capable of delivering full depth of book (Level II), Quotes (Level I) and Trades. It delivers public information in either a consolidated or non-consolidated form. It is capable of providing these feeds for all sources IRESS currently processes (market royalty dependent) and is built on top of the stable base of the existing IRESS Market Data infrastructure that has been feeding the majority of the professional market community in multiple countries for over the last decade.

IRESS Market Feed target usage is for areas that require low latency public market data such as for feeding trading algorithms, publishing data for display (WebSites, trading/market data applications etc.) and any place that needs a market wide streaming public data. It's main advantage over reading the market feed directly is that it is normalised (meaning one set of development to read X markets) and that it supports multiple variants of the original feeds including multi-market consolidated, individual market, delayed and end of day.

IRESS Market Feed features the following data elements (in binary format) publicly available for all exchange-listed equities securities traded via IRESS system:

Quotes	IRESS Market Feed provides comprehensive real-time market information for securities. Quote data includes price, bids, asks, volumes, and more. Quote lines can be either for a specific market or a 'merged' quote view across multiple markets.
Depth	IRESS provide full order depth of a security including Price/Volume/Order Number/Priority and publicly available broker numbers, using the standard IRESS Market Feed format. IRESS Market Feed uses a series of order messages to track the life of a customer order. IRESS Market Feed carries order level data for different exchanges.
Trades	IRESS provides full trade feeds across all supported trading destinations. Each trade is given a sequence number that is unique in the current trading session.

In addition, IRESS Market Feed also provides recovery features in case of disconnection from the feed for whatever reason.

Quote	IRESS Market Feed provides a current snapshot picture of the quote for each security code. Note: it does not provide a historical replay facility.
Depth	IRESS Market Feed provides a current snapshot picture of the Depth for each security code, so that only updates are needed after that. Note: it does not provide a historical replay facility.
Trade	IRESS Market Feed provides recovery within the scope of the most recent trading session from any point based on the trade sequence number. Using this facility, recovery of every missed trade is possible.

COMMUNICATION SEQUENCE

The IRESS Market Feed is made up of a series of messages. Each message has a predetermined packet length. The length is based on the message type. The messages that make up the IRESS Market Feed protocol are delivered using IRESS TCP Feed protocol that takes care of packeting. This protocol also gives the capability to skip over unknown or extended records without losing the stream. See Appendix 1: IRESS TCP Feed protocol to learn more about this protocol.

Each different data feed type is accessed through a separate TCP port (Quote: 10117, Depth: 10118 and Trade: 10119 unless advised otherwise). Multiple connections to different feeds are required if a combination of quote, depth, and trades are required.

Upon the connection, the client sends a Login Request Packet with user name and password as we as other necessary information.

The server verifies the user authentication information and reply to the client if the login attempt is a success or not, with either a Login Accepted Packet or a Login Reject Packet. For complete information about these packets, refer to relevant sections in Appendix.

If the log-in is successful, the client sends a subscription sequence to the server. It is possible for a client to subscribe to a single feed from a single market or it is be possible to subscribe to a feed from all markets (e.g. Subscribe to ASX@AUX and get a feed from CHI-X, ASX Trade Match and ASX Pure Match through a single subscription, getting a merged quote line and full detailed order book).

After the server receives the subscription request, it will confirm whether the subscription request is a success or not. If it is successful, for each security instrument belonging to the subscribed channels, the feed server will send an initial security information message, then a recovery sequence (i.e. for quote and depth a snapshot picture, for trade, a series of trade data messages since the specified recovery point), followed by an update stream in the form of data response messages of the specified feed type.

To disconnect from a feed server, the client simply closes the TCP connection.

DATA TYPES

- All integer fields are signed little-endian binary encoded numbers.
- Price fields are represented as IEEE 754 floating point in double precision format. Prices are in Dollars where applicable.
- Float fields are IEEE 754 floating point value in double precision format.
- All String fields are left justified and terminated with Null (Zero).
- Char fields are composed of one-byte characters.
- A Datetime field is an 8 byte little-endian integer that represents 100-nanosecond intervals since January 1st 1601. All Datetime values are local to the market. Use the GMT/UTC offset field to translate to other time zones. Times are only accurate to the precision supplied by the markets. IRESS generated times may be substituted on a message where the market supplied time is not supplied or processed by our feeds.
- All Message Type fields are represented by two characters. NOTE: the message type is not a NULL-terminated string and may use a blank space as the second character.
- All messages sent between the servers and the clients use a structure as below:

Data Message				
Name	Offset	Len	Value	Notes
Message Type	0	2	Char	Identifying the message type
Message Content		Variable	Any	Defined by each message type.

These messages are either carried by clients wrapped in Control Data Packets, or by servers wrapped in Feed Data Packet.

CLIENT REQUEST MESSAGES

After login, the client sends a subscription request sequence to the feed server to specify the required subscriptions. The subscription sequence consists of a Subscription Request Message and optionally a number of subscription supplement messages such as trade recovery request messages. This subscription request sequence tells the feed server the data source the client is interested in, and from which point to recover.

The subscription request sequence can be sent multiple times in a session which enables the clients to subscribe to different data source simultaneously. The client can only initiate a new subscription request sequence after current sequence is completed, i.e. all supplement packets have been sent.

The subscription request sequence provides a way for the clients customize the feed stream. These messages must be packed in an IRESS TCP Feed Control Data Packet (Packet type = 'U').

In response to a subscription request sequence, the server sends back a Subscription Confirmation Message..

SUBSCRIPTION REQUEST MESSAGES

Message Type = 'SU'

A Subscription request message is sent by a connected client to specify from which exchange, data source and data source board the client wants the data from. A client can send out multiple subscription request messages to request data from more than one source.

Subscription Request Message				
Name	Offset	Len	Value	Notes
Exchange		17	String	A code representing the exchange.
Data Source		9	String	A code representing the data source.
Data Source Board		9	String	A code representing the data source board.
Subscription Supplement Packet Count		4	Integer	A number indicating the number of optional subscription supplement packets such as Trade Recover Request messages immediately following this message. At this stage, only trade recovery makes use of this field. -1 – Full recovery of current server session 0 – No recovery >0 - The number of optional subscription supplement packets such as Trade Recover Request messages immediately following this message.

TRADE RECOVERY REQUEST MESSAGES

Message Type = 'TR'

A trade recovery request message specifies the largest trade number the client has ever received for a specified security within a session. Typically a client uses a sequence of trade recovery request messages following a Subscription Request Message to notify the feed server from which point to recover for a security. The number of the messages within the sequence is defined by Subscription Supplement Packet Count field in the Subscription Request Message above.

Trade Recovery Request Message				
Name	Offset	Len	Value	Notes
Security Id		4	Integer	The unique security ID, which was assigned to the security in a security information message previously.
Session Id		2	Integer	A session id identifying the session that the recovery point belongs to.
Recovery Point		4	Integer	The largest trade number the client has ever received for this particular security.

GENERAL INFORMATIONAL MESSAGES

General Informational Messages are sent by the data feed server to the clients in order to create or update information about an exchange, data source or a data source board. These messages are wrapped as the payload in a Feed Data Packet defined in IRESS TCP Feed protocol.

SUBSCRIPTION CONFIRMATION MESSAGE

Message Type = 'SC'

This message is the server's response to the Subscription Request Messages sent from a client.

Subscription Confirmation Message				
Name	Offset	Len	Value	Notes
Exchange		17	String	A code representing the exchange.
Data Source		9	String	A code representing the data source.
Data Source Board		9	String	A code representing the data source board.
Error Code`		2	Integer	A number representing the request result. 0 = Success. -1 = Undefined failure. -2 = The number of received recover requests does not match with the number defined in subscription request message after a certain period of time. -3 = Server not ready. Try again later. -4 = The specified exchange/datasource/board is not supported. -5 = the user has no permission to access specified exchange/datasource/board.

EXCHANGE INFORMATION MESSAGE

Message Type = 'XX'

Throughout a session, the server sends out Exchange Information Messages to update the status of an exchange-data source pair.

Exchange Information Message Content				
Name	Offset	Len	Value	Notes
Exchange		17	String	A code representing the exchange.
Data Source		9	String	A code representing the data source.
Status		51	String	The status of the exchange-data source.
Exchange Time		8	Datetime	The time of the exchange-data source.
Error Code		2	Integer	Error with the exchange-data source. 0 – OK -1 – Exchange is down. Data will resume once it is back up and running.

DATA SOURCE INFORMATION MESSAGE

Message Type = 'XD'

Data Source Information Message describes a data source associated with an exchange, and assigns an ID to this data source, unique in this exchange, so that future security information can refer to this ID to identify the data source.

Exchange Information Message Content				
Name	Offset	Len	Value	Notes
Exchange		17	String	The exchange associated with the data source.
Data Source		9	String	The name of the data source.
Data Source ID		2	Integer	The id unique to this exchange. Starting from 1.

Message may be sent throughout the session whenever a data source comes up from the update stream.

DATA SOURCE BOARD INFORMATION MESSAGE

Message Type = 'XB'

The Data Source Board Information Message is used to describe a data source board associated with a particular exchange and data source. This id is unique in this data source along with the exchange. The data feed response message refers to this ID to identify its data source board.

Data Source Board Information Message Content				
Name	Offset	Len	Value	Notes
Exchange		17	String	The exchange this data source is associated to.
Data Source ID		2	Integer	The data source id this board belongs to.
Data Source Board		9	String	The name of the data source board.
Data Source Board ID		2	Integer	The id unique to this data source board. Starting from 1.

This message may be sent by the server throughout the session whenever a data source board comes up from the update stream.

Based on the three messages above, the client will be able to build up a series of mapping between exchanges, data sources, and data source boards.

SECURITY INFORMATION MESSAGE

Message Type = 'IM'

Every security is assigned a security id, which will be used to identify a security in all data feed response messages. This id is unique for a security but may change between different login sessions and hence must not be cached between sessions.

Security Information Message Content				
Name	Offset	Len	Value	Notes
Security Id		4	Integer	A unique ID.
Security Code		33	String	The security code.
Exchange		17	String	A code representing the exchange.

This message may be sent by the server throughout the session whenever:

- A new security is encountered in the update stream.
- Any relevant information of a security has been changed.
- A new security is added.

DATA RESPONSE MESSAGES

As the response data stream, different type of data feed server serves different type of feed streams: Quote, or Depth, or Trade. These messages are also wrapped as the payload in a Feed Data Packet defined in IRESS TCP Feed protocol.

DATA RESPONSE MESSAGE STRUCTURE

Following the message type, the content of each feed response message consists of two parts as illustrated below:

Data Response Messages				
Name	Offset	Len	Value	Notes
Data Response Message Header		18		See below.
Data Response Message Body		Variable	Any	Defined by each message type.

DATA RESPONSE MESSAGE HEADER

Data Response Message Header					
Name		Offset	Len	Value	Notes
Instrument Set	Security Id		4	Integer	The unique security ID, which was assigned to the security in the security information message previously.
	Data Source Id		2	Integer	The data source ID, which was assigned in Data Source Information Message. 0 means data source undefined.
	Data Source Board Id		2	Integer	The data source board ID, which was assigned in Data Source Board Information Message. 0 means board undefined.
Update Time			8	Datetime	Update time, only accurate to the time provided by the exchange or internal system when no timestamp is provided for the accompanying message.
Session Id			2	Integer	A session id indicating the session this data message belongs to. This session id can be used later in recovery.

Data Response Message Bodies include following types.

RECOVERY SEQUENCE MESSAGES

The recovery sequence is sent upon the subscription. Each security has a corresponding recovery sequence. The sequence may contain zero data message. The beginning of a recovery sequence is indicated by a Begin Of Recovery message, followed by a series of data response messages, and in the end an End Of Recovery message indicating the end of the sequence.

BEGIN OF RECOVERY MESSAGE

Message Type = 'BR'

The body of this message type is empty.

BEGIN OF RECOVERY MESSAGE BODY				
Name	Offset	Len	Value	Notes

END OF RECOVERY MESSAGE

Message Type = 'ER'

The body of this message type is empty.

END OF RECOVERY MESSAGE BODY				
Name	Offset	Len	Value	Notes

QUOTE MESSAGES

STATUS MESSAGE

Message Type = 'QS'

QUOTE MESSAGE BODY – Status				
Name	Offset	Len	Value	Notes
Security Status		1	Char	The status of security, the possible values and respective meanings are specific to different exchanges. Refer to each exchange's specification for the details.

BID MESSAGE

Message Type = 'QB'

QUOTE MESSAGE BODY – Bid				
Name	Offset	Len	Value	Notes
Bid Price		8	Price	
Bid Num		4	Integer	The number of orders in the bid at this price.
Bid Volume		8	Float	
Bid Data Source		9	String	For merged quotes only.

ASK MESSAGE

Message Type = 'QA'

QUOTE MESSAGE BODY – Ask				
Name	Offset	Len	Value	Notes
Ask Price		8	Price	
Ask Num		4	Integer	The number of orders in the ask at this price.
Ask Volume		8	Float	
Ask Data Source		9	String	For merged quotes only.

TRADE MESSAGE

Message Type = 'QT'

QUOTE MESSAGE BODY – Trade				
Name	Offset	Len	Value	Notes
Last Price		8	Price	Last price of the security.
Number Of Trades		4	Integer	Number of trades that have occurred so far on the day.
Trade Time		8	Time	
Market Value		8	Float	For on market trades only.
Market Volume		8	Float	For on market trades only.
Cum Value		8	Float	
Cum Volume		8	Float	

QUOTE MESSAGES CONTINUED

MATCH MESSAGE

Message Type = 'QM'

QUOTE MESSAGE BODY – Match				
Name	Offset	Len	Value	Notes
Match Volume		8	Float	
Surplus Volume		8	Float	
Indicative Price		8	Price	

FULL MESSAGE

Message Type = 'QF'

Quote Full Message are sent as the first snapshot message, and later on when the changes are more than any other type of quote messages can contain and hence only a full message will do.

QUOTE MESSAGE BODY – Full				
Name	Offset	Len	Value	Notes
GMT Timezone		2	Integer	GMT Timezone offset in minutes.
Quotation Basis		12	String	Quotation basis.

QUOTE MESSAGE BODY – Full				
Security Type		4	Integer	This indicates the type of a security. See Security Types for details
Open Price		8	Price	Opening price.
High Price		8	Price	Today's high.
Low Price		8	Price	Today's low.
Close Price		8	Price	Close price.
Status Notes		9	String	They are provided by the exchange and hence exchange-specific.
Security Status		1	Char	The status of the security on the exchange. See Security Status for details.
Bid Price		8	Price	
Bid Num		4	Integer	
Bid Volume		8	Float	
Bid Data Source		9	String	For merged quotes only.
Ask Price		8	Price	
Ask Num		4	Integer	
Ask Volume		8	Float	
Ask Data Source		9	String	For merged quotes only.
Last Price		8	Price	Last price of the security.
Number Of Trades		4	Integer	Number of trades that have occurred so far on the day.
Trade Time		8	Time	
Market Value		8	Float	For on market trades only.
Market Volume		8	Float	For on market trades only.
Cum Value		8	Float	
Cum Volume		8	Float	
Match Volume		8	Float	
Surplus Volume		8	Float	
Indicative Price		8	Price	

DEPTH MESSAGES

Any price modifications to an order will always invoke a Delete and an Add action. The Order No does not change with these modifications.

Sortkey and SortSubkey are used as a universal coordinate system to sort active orders in the depth stack. They allow priority of orders to be easily identified across multiple markets. They are used in pairs (SortKey, SortSubkey) and sorting of orders is achieved by sorting each orders by their SortKey followed by their SubSortKey in an ascending order. If you are maintaining a depth stack you'll have to use these keys.

SINGLE ORDER MESSAGE

Message Type = 'DS'

SINGLE DEPTH ORDER MESSAGE BODY				
Name	Offset	Len	Value	Notes

BidOrAsk		1	Char	B – Bid A – Ask.
Sort Key		8	Integer	Part of the order's [SortKey, SubSortKey] key.
Sort SubKey		4	Integer	Part of the order's [SortKey, SubSortKey] key.
Action		1	Char	A code representing the action to be taken on the order. A – Add D – Delete M - Modify
Price		8	Price	
Volume		8	Float	
Order Type		4	Integer	A bitwise value that indicates properties associated with the order. See Depth Order types for the details.
Broker No		2	Integer	Only available on public market orders. Zero is used if this information is not available.
Order No		8	Integer	

CONSOLIDATED ORDER MESSAGE

Message Type = 'DC'

Consolidated depth order message are only generated for ETFs, Options, Warrants and Futures where the depth view is limited to a consolidated view. For equities (e.g. ASX@TM), you should only receive single order messages.

CONSOLIDATED DEPTH ORDER MESSAGE BODY				
Name	Offset	Len	Value	Notes
BidOrAsk		1	Char	B – Bid A – Ask
Sort Key		8	Integer	Part of the order's [SortKey, SubSortKey] key.
Sort SubKey		4	Integer	Part of the order's [SortKey, SubSortKey] key.
Action		1	Char	A code representing the action to be taken on the order. A – Add D – Delete M – Modify
Price		8	Price	
Volume		8	Float	
Order Type		4	Integer	A bitwise value that indicates properties associated with the order. The possible values and respective meanings are specific to different exchanges. Refer to each exchange's specification for the details.
Order Count		4	Integer	Number of orders at this price level. For consolidated orders only.

DEPTH CLEAR MESSAGE

Message Type = 'DE'

The message body is empty.

DEPTH CLEAR MESSAGE BODY				
Name	Offset	Len	Value	Notes

SORTKEY UPDATE MESSAGE

Message Type = 'DO'

This message is used to update a [SortKey, SortSubkey] key. Sometimes this message will be followed by an order message with a modify action to modify other order fields as well.

DEPTH KEY UPDATE MESSAGE BODY				
Name	Offset	Len	Value	Notes
BidOrAsk		1	Char	B – Bid
'A' – Ask.				
Sort Key		8	Integer	Part of the order's [SortKey, SubSortKey] key.
Sort SubKey		4	Integer	Part of the order's [SortKey, SubSortKey] key.
New Sort Key		8	Integer	The new key value.
New Sort SubKey		4	Integer	The new key value.

DEPTH MESSAGES CONTINUED

BULK UPDATE MESSAGE

Message Type = 'DU'

Bulk update messages are useful to updating prices and/or order types of multiple orders at one go.

DEPTH BULK UPDATE MESSAGE BODY				
Name	Offset	Len	Value	Notes
BidOrAsk		1	Char	B – Bid A – Ask
Begin Sort Key		8	Integer	The beginning of the sort key range.
Begin Sort SubKey		4	Integer	The beginning of the sort key range.
End Sort Key		8	Integer	The end of the sort key range.
End Sort SubKey		4	Integer	The end of the sort key range.
Action		1	Char	S – Set the field specified by Field Flag within the specified sort key range. I – Set a bit in a field. The value should be applied via a bitwise OR operator to the current values of the field specified by the Field Flag for all orders within the sort key range. J – Unset a bit in a field. The value should be applied via an inverse (NOT) of the value followed by a bitwise AND operator to the current values of the field specified by the Field Flag for all orders within the sort key range. E – Delete rows within the sort key range. The Field Flag is not used in this instance
Field Flag		1	Char	P – Updates to be applied to the Price field. O – Updates to be applied to the Order Type field. '' – Field update values are not applicable.
Price		8	Price	NOTE: This value is only applied when Field Flag is set to 'P'.
Order Type		4	Integer	A bitwise value that indicates properties associated with the order. The possible values and respective meanings are specific to different exchanges. Refer to each exchange's specification for the details. NOTE: This value is only applied when Field Flag is set to 'O'.

TRADE MESSAGES

TRADE MESSAGE

Message Type = 'TR'

This message is sent whenever a new trade occurs or a trade is updated.

The client should keep track of the largest trade number it has received as this record can be used for checking whether this message is a new trade or an update. This number is used for recovery when needed. See Trade Recovery Request Message for more information about using the trade number in recovery.

When receiving a Trade Message, the client must check the trade number against its trade database. If the number is larger than any previous trade number, then the Trade Message is for a new trade. If the trade number is smaller than the largest trade number, the existing trade in the trade database must be updated with the information in the Trade Message. If the trade number does not exist in the trade database, a new trade can be created for this information.

TRADE MESSAGE BODY				
Name	Offset	Len	Value	Notes
Seller Id		2	Integer	Only available on public market orders.
Seller Order Id		8	Integer	
Buyer Id		2	Integer	Only available on public market orders.
Buyer order Id		8	Integer	
Trade No		4	Integer	The unique trade number assigned to this trade.
Trade Value		8	Float	
Trade Volume		8	Float	
Trade Price		8	Price	
Trade Time		8	Datetime	The timestamp of the trade.
Action Flag		4	Integer	A bitwise value that indicates properties associated with the trade. See Trade Action Flags for details.
Condition Codes		9	String	A list of character sets which represent conditions of the trade. See Trade Condition Codes for the details.

TRADE MESSAGES CONTINUED

TRADE CANCEL MESSAGE

Message Type = 'TC'

This message is sent to cancel an existing trade. It contains full information about the trade being cancelled.

It also contains an extra trade number which is assigned to the cancelling operation itself. This number cannot be used for trade checking or recovery purposes.

TRADE CANCEL MESSAGE BODY				
Name	Offset	Len	Value	Notes
Seller Id		2	Integer	Only available on public market orders.
Seller Order Id		8	Integer	
Buyer Id		2	Integer	Only available on public market orders.
Buyer order Id		8	Integer	
Trade No		4	Integer	The unique trade no assigned to the trade.
Trade Value		8	Float	
Trade Volume		8	Float	
Trade Price		8	Price	
Trade Time		8	Datetime	The timestamp of the trade.
Action Flag		4	Integer	A bitwise value that indicates properties associated with the trade. See Trade message.
Condition Codes		9	String	A list of character sets which represent conditions of the trade.
Cancel Trade No		4	Integer	The trade number of the cancelling operation itself.

BROKER UPDATE MESSAGE

Message Type = 'TB'

This message is sent to update the broker information of an existing trade.

When receiving a Broker Update Message, the client should check if it has received the trade with specified trade number. If the trade number has been received, the trade should be updated with the broker information. If the trade number has not been received, the message is ignored.

The trade number cannot be used for trade checking or recovery purposes.

TRADE MESSAGE BODY				
Name	Offset	Len	Value	Notes
BidOrAsk		1	Char	B – Buyer A – Seller
Broker Id		2	Integer	Only available on public market orders.
Trade No		4	Integer	
Trade Time		8	Datetime	The timestamp of the trade.

APPENDIX 1: IRESS TCP FEED PROTOCOL

OVERVIEW

IRESS TCP Feed is a lightweight point-to-point protocol, built on top of TCP/IP sockets that allow delivery of a set of messages from a server to a client in real-time. IRESS TCP Feed guarantees that the client receives each message generated by the server, even if the messages grow or unknown messages are received.

IRESS TCP Feed clients can send messages to the server.

IRESS TCP Feed is designed to be used in conjunction with our higher lever protocol that specify the contents of the messages that IRESS TCP Feed messages deliver. The IRESS TCP Feed protocol layer is opaque to the higher-level messages. Note that messages may include any possibly byte.

IRESS TCP Feed also includes a scheme that allows the server to authenticate the client on login.

IRESS TCP FEED LOGICAL PACKETS

The IRESS TCP Feed client and server communicate by exchanging a series of logical packets. Each IRESS TCP Feed logical packet has:

1. a two byte little-endian length that indicates the length of rest of the packet (meaning the length of the payload plus the length of the packet type – which is 1)
2. a single byte header which indicates the packet type
3. a variable length payload (can be zero-sized).

IRESS TCP Feed Logical Packet Structure

Two Byte Packet Length	Packet Type	Payload

Notes:

- The IRESS TCP Feed logical packets do not necessarily map directly to physical packets on the underlying network socket; they may be broken apart or aggregated by the TCP/IP stack.
- The IRESS TCP Feed protocol defines a maximum payload length of 2^16-3 bytes.
- The payload may contain any binary sequence.
- This design gives Iress Market Feed clients the capability to safely ignore a packet type that they do not understand. It means an Iress Market Feed server can introduce new features (e.g. new types of packets) while keeps the backward-compatibility. An Iress Market Feed client should always ignore the packet types it has no knowledge of.

PROTOCOL FLOW

An IRESS TCP Feed connection begins with the client opening a TCP/IP socket to the server and sending a Login Request Packet. If the login request is valid, the server responds with a Login Accepted Packet and begins sending logical packets. The connection continues until the TCP/IP socket is broken.

Each packet carries a single higher-level protocol message.

IRESS TCP Feed also permits the client to send messages to the server using Control Data Packets at any time after the Login Accepted Packet is received.

LOGICAL PACKETS FROM EITHER SIDE

These packets can be sent by either side of an IRESS TCP Feed connection.

DEBUG PACKET

A debug packet can be sent by either side of an IRESS TCP Feed connection at any time. Debug packets are intended to provide human readable text that may aid in debugging problems. Debug Packets should be ignored by both client and server application software.

Debug Packet				
Name	Offset	Len	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet.
Packet Type		1	'+'	Debug Packet
Text		Variable	String	Free form human readable text, the length can be deducted from packet length.

HEARTBEAT PACKET

A Heartbeat Packet must be sent by either sides when more than 1 second passes where no data has been sent to the other side. One end can assume that the link is lost if it does not receive anything from the other end for 2 seconds or longer.

Heartbeat Packet				
Name	Offset	Len	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet.
Packet Type		1	'H'	Heartbeat Packet.

LOGICAL PACKETS FROM IRESS TCP FEED CLIENTS

LOGIN REQUEST PACKET

The IRESS TCP Feed client must send a Login Request Packet immediately upon establishing a new TCP/IP socket connection to the server.

Client and server must have mutually agreed upon the username and password fields. They provide simple authentication to prevent a client from inadvertently connecting to the wrong server. Both Username and Password are case-insensitive and should be padded on the right with spaces. The server can terminate an incoming TCP/IP socket if it does not receive a Login Request Packet within a reasonable period of time (typically 30 seconds).

Login Request Packet				
Name	Offset	Len	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet
Packet Type		1	'L'	Login Request Packet
Username		7	String	Username
Password		129	String	Password
Request Flag		2	Integer	The Iress Market Feed specification version. The HIBYTE is the major version, and the

				LOBYTE is the minor version. E.g. HIBYTE(RequestFlag) = 1, LOBYTE(RequestFlag) = 2 means the client supports Iress Market Feed Spec 1.2. A client sending Zero in this field will be treated as it conforms to Spec 1.1.

LOGICAL PACKETS FROM IRESS TCP FEED CLIENTS CONTINUED

CONTROL DATA PACKETS

The Control Data Packets act as an envelope to carry the control messages such as Subscription Request Messages that are transferred from clients to a server.

Control Data Packet				
Name	Offset	Len	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet.
Packet Type		1	'U'	
Message Type		2	Char	The message type.
Message Content		Variable		The actual message content, defined by a higher-level protocol.

LOGICAL PACKETS FROM IRESS TCP FEED SERVERS

SERVER INFORMATION PACKET

Upon the connection from a client, the server will immediately send a server information packet to the client to present brief information about the server.

Server Information Packet				
Name	Offset	Len	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet.
Packet Type		1	'I'	Server Information Packet
Server Name		21	String	
Server Type		1	Char	'Q' – Quote Feed Server 'D' – Depth feed server 'T' – Trade feed server
Server Version		9	String	A string representing the Iress Market Feed Spec version that this server supports. For example, if the server supports spec 1.2, then this string will be "1.2" (without the quotation marks). Note Iress Market Feed spec is kept backward-compatible, i.e. a higher-version of Iress Market Feed server supports a lower- version of clients. For example, a 1.2 Iress Market Feed server will support a 1.1 Iress Market Feed client.

LOGIN ACCEPTED PACKET

The IRESS TCP Feed server sends a Login Accepted Packet in response to receiving a valid Login Request from the client. This packet will always be the first non-debug packet sent by the server after a successful login request.

Login Accepted Packet				
Name	Offset	Len	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet.
Packet Type		1	'A'	Login Accepted Packet

LOGIN REJECTED PACKET

The IRESS TCP Feed server sends this packet in response to an invalid Login Request Packet from the client. The server closes the socket connection after sending the Login Reject Packet. The Login Rejected Packet will be the only non-debug packet sent by the server in the case of an unsuccessful login attempt.

Login Reject Packet				
Name	Offset	Len	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next. packet
Packet Type		1	'J'	Login Rejected Packet
Reject Reason Code		1	Char	See Login Reject Codes below.

Login Reject Codes	
Code	Explanation
A	Not Authorized. There was an invalid username and password combination in the Login Request Message.
F	The server is not ready. Try again later.
D	The user account is locked or disabled.
L	The login limit per this user has been reached. No more logins allowed until at least one of other logi ns closes.
U	Unspecified error. Try later or contact the system admin.

FEED DATA PACKET

The Feed Data Packets act as an envelope to carry the actual feed data messages that are transferred from feed servers to clients. Each Feed Data Packet carries one message from the higher-lever protocol. This type of packets makes up of the update stream of a server.

Since IRESS TCP Feed logical packets are carried via TCP/IP sockets, the only way logical packets can be lost is in the event of a TCP/IP socket connection failure.

Feed Data Packet				
Name	Offset	Len	Value	Notes
Packet Length	0	2	Integer	Number of bytes after this field until the next packet.
Packet Type		1	'D'	Feed Data Packet.
Message Type		2	Char	The message type.
Message Content		Variable		The actual message content, defined by a higher-level protocol.

APPENDIX 2: TYPES AND FLAGS

SECURITY TYPES

Note not all values are applicable to a certain market.

Security Type	
Type	Description
100	Equities
101	Rights Issues
102	Bonus Shares
103	New Shares
104	Trust Companies
105	Other Entitlements
106	Company Options
107	Warrants
108	Put Warrant
109	Special Call Warrants (instalment, Endowment, Basket, Capital Plus, Index)
110	Special Put Warrants (instalment, Endowment, Basket, Capital Plus, Index)
111	SGX Warrants
112	Exchange Traded Funds
113	Extended Settlement
120	Underlying ratio
121	Actual ratio contract
130	Security is in 'special market' trading
200	Preference Shares
205	Convertible Preference
300	Convertible Notes
301	Convertible Notes (Interest Rate Sec)
302	High Denomination Convertible Notes
501	Equity Call Options
502	Equity Put Options
503	LEPO
521	Equity Futures Call Option
522	Equity Futures Put Option
529	Index Futures Call Option
530	Index Futures Put Option
533	Wheat Future Call Option
534	Wheat Future Put Option
541	Index Call Option

542	Index Put Option

SECURITY STATUS

The values of Security Status appearing in a Quote Full message are specific to different exchanges. Refer to each exchange's specification for the details.

For reference purpose, below lists the definitions from ASX.

Security Status (ASX)	
Status	Description
	Open
A	Adjust - orders can be modified, but no trades processed
C	Closing
E	Inquiry only - no orders can be entered or modified
F	FAST market (SFE only)
H	Trading Halt
I	Inquiry
J	After hours adjust
M	Market special
N	Pre Noticed Received
O	Opening - orders entered and modified but no trades executed
P	Pre-Open - orders entered and modified but no trades executed
S	Suspended
U	PRE_CSPA - orders entered and modified but no trades executed
W	Wait

DEPTH ORDER TYPES

Depth Order Type is a bitwise value that indicates properties associated with the order. Below lists all possible types. Note not all values are applicable to a certain market.

Depth Order Type	
Bitmask	Description
0x00000001	Short Sell
0x00000002	Market Bid
0x00000004	Price Stabilisation
0x00000008	Possible Iceberg
0x00000010	Undisclosed
0x00000020	Bait
0x00000040	Anonymous

0x00000080	MOC
0x00000100	All or None
0x00000200	Short Exempt
0x00000400	Minimum Guaranteed Fill
0x00000800	Non Resident
0x00001000	Market To Limit
0x00002000	Must Be Filled
0x00004000	Matched
0x00008000	Partial Matched
0x00010000	Possibly Matched
0x00020000	Frozen
0x00040000	On Stop
0x00080000	COP MBF
0x00100000	COP MKT
0x00200000	Post Only
0x00400000	Phantom

TRADE ACTION FLAGS

A bitwise value that indicates properties associated with the trade. Note not all values are applicable to a certain market.

Trade Action Flag	
Bitmask	Description
0x00000001	This trade affects the Open, High, Low, and Last prices.
0x00000002	This trade affect the market volume and value. Without this flag set, the trade is off-market.
0x00000004	This trade contains the accumulation of all traded volume of current session.
0x00000008	Reserved
0x00000010	Reserved
0x00000020	This trade is a re-transmit of an existing trade. Ignore if same trade has been received previously.
0x00000040	This is a trade cancelling operation. Only trade cancelling message carries this flag.
0x00000080	Reserved
0x00000100	The bid side is the aggressor
0x00000200	The ask side is the aggressor
0x00000800	This trade is an anonymous bid
0x00001000	This trade is an anonymous ask
0x00002000	This trade affects only the Open, High, and Low prices, not the Last.

0x00004000	Reserved.

TRADE CONDITION CODES

The values of Trade Condition Codes appearing in Trade messages are specific to different exchanges. Refer to each exchange's specification for the details.

For reference purpose, below lists the definitions from ASX.

Trade Condition Codes (ASX)	
Code	Description
BK	Buy Back
BV	Book Value
BW	Buy and Write
LN	Loan
ON	Overnight
OS	Overseas
SP	Special Crossing
NX	Trade At Or Within Spread
LR	Loan Return
PO	Permitted Trade During Post-Trading Hours Period
EP	Exercise Put
OR	Overseas Resident
PC	Permitted Trade During Pre-Trading Hours Period
SH	Short Sell
SX	Portfolio Special Crossing
SO	Special Crossing Other
XT	Crossing
ST	Price Stabilisation
BP	Booking
MI	Market Info
EC	Exercise Call
DR	Directed Reporting
SA	Special Crossing Sale
FD	Forward Delivery
FM	Foreign Market Sale
IB	Index Portfolio Replicating Special
CT	Combination Trade
LT	Late Trade

ET	Exchange Traded Funds Special
PT	Put-Through Trade
PR	Prompt Rebook
AD	ASX Disclosed
BB	Bulletin Board Trade
GL	Non-Scr Traded Govt Instr.
TM	Tailor Made Combination
WH	Non-Scr Traded Wholesale Instr.
EQ	Equity/Option Combination
CM	Tailor Made Combo 1 side
CP	Centre Point Priority Crossing
L1	Late Trade Report - Book Squaring
L2	Late Trade Report - Hedging Trades
L3	Late Trade Report - Order Completion
L4	Late Trade Report - Error Rectification
L5	Late Trade Report - Put Through
NG	Negotiated Deal
S1	Special Crossing = T1
P1	Put-Through Special Crossing = T2
P2	Put-Through Special Crossing = T3 < T2
DE	Delayed Reporting
CX	Centre Point
VM	Volume Match
VW	VWAP Trade