

Trayport Joule Direct

API Conformance Requirements

For read-write applications

Introduction

Trayport Conformance Testing ensures that third party or in-house applications are correctly interacting with the Trayport software and platforms. Testing is carried out for all major upgrades of external applications, with the primary aim of ensuring users receive the highest quality experience and the integrity of the Trayport environment is not put at risk.

Upon successful completion of the test, Trayport will provide a conformance certificate detailing the outcome of the test.

Benefits of Conformance Testing

- Better serve market participants by ensuring optimal connectivity of external applications with Trayport software.
- Provide market participants with assurance and confidence that conforming applications behave as expected and perform functions in a known manner.
- Consistency of standards across all applications, while all new versions are tested to ensure highest standards are maintained.
- Reduce risk and cost by safeguarding the integrity (stability, security and reliability) of the Trayport environment for the benefit of all our clients.

The Trayport conformance test is designed to be thorough but time efficient, thus reducing client overheads. Testing is carried out in Trayport's dedicated conformance environment for all new and major upgrades for applications connecting to the Trayport API.

The conformance test process typically takes up to 1 hour to complete for each client application.

The test will be performed in collaboration with Trayport staff. The participant will be instructed to execute specified certification test cases and Trayport will verify that the relevant application fulfils the expected requirements and behaviour.

Requirements

If the application is to view market orders, it must be able to:

- Handle an empty order book
- View prices on both sides of the market
- Properly display price and quantity updates to own and market orders
- Handle the removal of prices from an order book
- Handle changes that occur to an order book while the application is disconnected
- Handle choice markets
- Handle crossed markets
- Handle multiple prices on the same level
- Handle negative prices
- Handle decimal quantities
- Handle venue implied prices
- Handle prices from multiple venues for the same contract
- Display the last traded prices correctly.

If the application is to view private orders, it must also be able to:

- Handle aggression (full or partial) of private orders
- Handle the insertion, update or removal of private orders by the same or other accounts at the same company
- Handle private orders with negative prices
- Handle private orders with decimal quantities

If the application is to view market trades, it must be able to:

- Handle the insertion, update, deletion and restoration of trades
- Handle manual trades
- Handle voice trades
- Handle trades that are not linked to any order but are not marked as manual trades.

If the application is to view private trades, it must also be able to:

- Display the counterparties to trades where provided
- Handle private trades that have not been provided with counterparty information.

If the application is to be able to manage orders, it must be able to:

- Enter orders
- Update the broker, price and quantity of an order
- Delete orders
- Enter an order that would create a choice market
- Enter an order that would cross the market
- Enter an order with a negative price
- Enter order with a decimal quantity
- Enter an order with a hidden quantity
- Update, delete, withhold or firm orders inserted by other users at the same company

- Handle its own orders being updated, withheld or removed by other users at the same company
- Handle its own orders being fully or partially aggressed
- Handle an order being instantly aggressed, updated or removed upon insertion
- Handle the following errors when attempting an order action:
- Insufficient instrument permissions at the venue
- Choice market not allowed
- Order would create an invalid spread
- Venue disconnected.

If the application is to be able to deal orders, it must be able to:

- Deal a specific order
- Deal a given volume (if this feature is required)
- Deal a given volume across multiple venues (if this feature is required)
- Partially deal a specific order.

In addition to the above, all applications must:

- Display their connection status to the end user
- Reconnect automatically if disconnected (this is usually done automatically by the API)
- Stop attempting to reconnect if the account is disabled
- Be able to handle changes to orders and trades that occurred while the application is disconnected
- Throttle actions to 100/sec or 1000/min
- Act in line with the regulatory requirements of the connected venues
- Store passwords securely
- Use the subscription method rather than repeated polling if real-time data access is required
- Have protection against feedback loops if using algorithmic trading.

We would require all of the above to be demonstrated as part of the conformance test, and would also test the application's response to many of the above actions being performed by a broker (for example, broker inserting, updating and removing orders, and deals on your behalf).

For any conformance related queries and to schedule the conformance test, please contact Trayport Support at support@trayport.com or +44 (0)20 7960 5555.