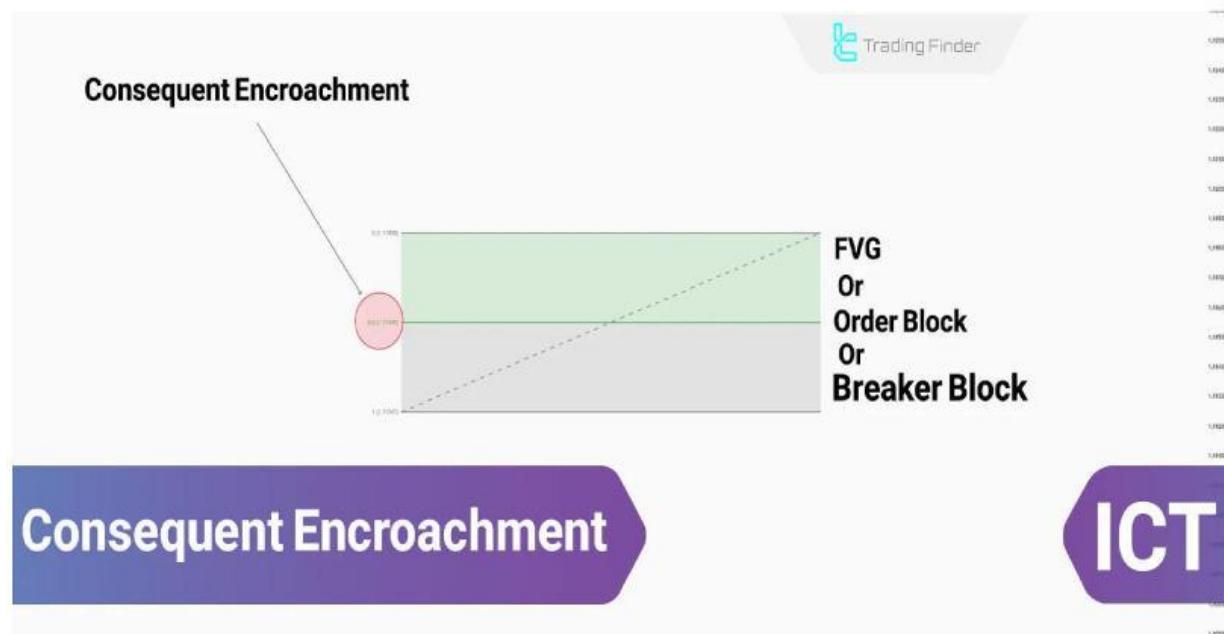


Consequent Encroachment in ICT Trading; Combining With FVG, OB, BB and PD Arrays

Consequent Encroachment (CE) focuses on a level where the price is highly likely to react. This concept is crucial, especially in analyzing **Fair Value Gaps (FVGs)** and other trading structures.

Traders can use this approach to identify **optimal entry** and **exit points** in the market.



A conceptual illustration of Consequent Encroachment, which can be integrated with FVG, OB, or BB

What is ICT Consequent Encroachment (CE)?

CE refers to the **50% level** of a trading structure, such as the midpoint of a **Fair Value Gap (FVG)** or other **ICT PD Arrays**. This level is considered the “**midpoint**” of a gap or a balanced price zone where the price is most likely to react.

Unlike full gap fills, CE allows traders in the **forex market** to take action before an FVG or **Order Block** is filled, offering an approach for anticipating price reversals or trend continuation.

Advantages and Disadvantages of CE

Like other concepts in **ICT style**, **Consequent Encroachment** has its own strengths and limitations:

Disadvantages	Advantages
Potential for false signals in lower timeframes	Identifies key market levels
Requires advanced experience for accurate usage	Improves risk management and capital efficiency
Price may ignore CE levels at times	Applicable across all markets and timeframes
Complex for beginner traders	Enhances trading precision

How to Identify Consequent Encroachment (CE)?

To identify **CE** in an FVG structure, follow these steps:

- 1. Locate the Fair Value Gap (FVG):** Identify a bullish or bearish **FVG** on the chart;
- 2. Use the Fibonacci Tool:** Apply Fibonacci from the highest point to the lowest point of the FVG;
- 3. Mark the 50% Level:** The **50% Fibonacci level** represents the **Consequent Encroachment**, which is an entry or exit point.



An example of a bearish Consequent Encroachment in a Fair Value Gap, identified with the Fibonacci tool

Below is an example of a **bullish CE**:



A bullish Consequent Encroachment example in a Fair Value Gap, identified with the Fibonacci tool

How Does CE Apply to Other ICT Concepts?

Apart from **Fair Value Gaps (FVGs)**, **Consequent Encroachment** is also applicable in the following **ICT structures**:

- ◆ **Order Blocks (OBs):** CE within an Order Block acts as a price equilibrium point;
- ◆ **Breaker Blocks (BBs):** CE helps identify critical levels in **Breaker Blocks**;
- ◆ **New Week Opening Gaps (NWOGs):** **Weekly opening gaps** contain a **CE level** that acts as a **liquidity magnet**.

Key Points for Identifying and Trading with CE

To trade using Conceptual Equilibrium (CE), combine it with **liquidity shifts** and **structural changes** and focus on **higher timeframes** for greater reliability

- ◆ **Combining with Other Tools:** CE works best when integrated with **other trading concepts**;
- ◆ **Higher Timeframes:** **CE levels** are more reliable in higher timeframes as they filter out the noise;
- ◆ **Risk Management:** **CE alone is insufficient** for trade execution; combining it with **other ICT elements** is essential to avoid losses.

Practical Example of Trading with Consequent Encroachment

Below is a **real-world example** demonstrating how price reacts to **CE in an uptrend** on USD/JPY chart, where price reaches **Consequent Encroachment** and reverses downward.



A real trade example showing the price reaching the midpoint of an FVG (CE level) before reversing on USD/JPY chart

Conclusion

The **ICT Consequent Encroachment (CE)** is a **refined tool** for optimizing **price zones** in ICT structures such as **FVGs and Order Blocks**. By dividing key market areas into precise sections, CE enhances **entry accuracy and trade execution**.

source:

1.our website link :

<https://tradingfinder.com/education/forex/ict-consequent-encroachment/>

2.all Education :

<https://tradingfinder.com/education/forex/>

3.TradingFinder Support Team (Telegram):

<https://t.me/TFLABS>



[Educational link](#)



[tradingfindercom](#)