

Running robots on virtual hosting is easy

Follow our step-by-step MetaTrader VPS guide for beginners



Your article is at the Draft stage.

You can freely edit it on your own: change sections, edit text and images, correct errors. Saving a draft creates a separate independent version of article. If you wish, you can access any previous version. As soon as (in your opinion) your article is ready for review, you can send it for moderator's check using the **Send for review** button.

Send for review

Development of trading strategies

[edit](#)

[Mduduzi Gift Ngcebo Langa](#) | 29 April, 2025

A trading strategy is a predefined set of rules for entering and exiting positions in financial markets.

Development of Trading Strategies in MetaTrader 5 (MQL5) Introduction

In the fast-paced world of financial markets, algorithmic trading has become a key advantage for both retail and institutional traders. Platforms like MetaTrader 5 (MT5) and the MQL5 language provide an environment where robust, automated trading strategies can be created, tested, and deployed with high precision. This article delves into the complete lifecycle of developing a trading strategy in MQL5—from concept to code, testing, and optimization.

1. Understanding Trading Strategies

1.1 What is a Trading Strategy?

A trading strategy is a predefined set of rules and conditions that determine when to buy or sell a financial asset. These conditions are typically based on:

- Technical indicators (e.g., RSI, Moving Averages)
- Price action (e.g., support/resistance levels)
- Timeframes (e.g., only during London or New York session)
- Market sentiment or volatility filters

1.2 Types of Strategies

Strategies vary depending on the trader's objectives:

- **Trend-Following:** Riding the market momentum using indicators like EMA or MACD.
- **Counter-Trend / Mean Reversion:** Betting against overextensions using Bollinger Bands, RSI, or CCI.

- **Breakout Strategies:** Trading price explosions from consolidation ranges or key levels.
 - **Session-Based:** Executing trades based on session behaviors (London Open, NY Close, Asian session).
-

2. Strategy Development Process

2.1 Define the Hypothesis

Every good strategy begins with an idea. Examples:

- “Gold tends to break out during the London open with high volatility.”
- “After a strong Monday on NASDAQ, Tuesday often sees a pullback.”

Formulate clear assumptions and begin with data analysis or visual chart inspection.

2.2 Translate to Rules

Turn your hypothesis into algorithmic logic. Example for a trend strategy:

- If 50 EMA is sloping upward
- RSI(14) is above 55
- Price closes above the 50 EMA
- Trade only between 09:00-12:00 server time

These rules form the core of your EA (Expert Advisor).

2.3 Choose Indicators

Select indicators and parameters wisely:

- **Trend confirmation:** EMA, MACD, ADX
- **Entry/exit triggers:** RSI, Stochastic, Bollinger Bands
- **Volatility filters:** ATR, volume, candle range
- **Risk filters:** Max spread, slippage, news time

2.4 Time and Symbol Selection

Each strategy performs differently on various instruments and timeframes. Scalping on M5 may work on US30, but fail on EURUSD. Consider:

- **Asset:** Forex, commodities, indices, crypto
 - **Timeframe:** M1, M15, H1, H4, Daily
-

3. MQL5 Coding: From Logic to Execution

3.1 Creating the Framework

Start with the EA template:

```
#include <Trade\Trade.mqh>
CTrade trade;

input int EMA_Period = 50;
input int RSI_Period = 14;
input double RSI_Threshold = 55.0;
input double SL_Points = 100;
input double TP_Points = 150;
input double RiskPercent = 2.0;
```

3.2 Entry Logic

```
void OnTick()
{
    if (PositionsTotal() > 0) return;

    double ema = iMA(NULL, 0, EMA_Period, 0, MODE_EMA, PRICE_CLOSE, 0);
    double rsi = iRSI(NULL, 0, RSI_Period, PRICE_CLOSE, 0);
    double price = Close[0];

    datetime time_now = TimeCurrent();
    int hour = TimeHour(time_now);
    if (hour < 9 || hour > 12) return;

    if (price > ema && rsi > RSI_Threshold)
    {
        double lot = CalculateLotSize(SL_Points);
        trade.Buy(lot, _Symbol, price, price - SL_Points * _Point, price + TP_Points * _P
    }
}
```

3.3 Lot Size Calculation

```
double CalculateLotSize(double sl_points)
{
    double risk = AccountBalance() * RiskPercent / 100.0;
    double tick_value = SymbolInfoDouble(_Symbol, SYMBOL_TRADE_TICK_VALUE);
    double lot = risk / (sl_points * _Point * tick_value);
    return NormalizeDouble(lot, 2);
}
```

4. Testing and Optimization

4.1 Backtesting in Strategy Tester

1. Open Strategy Tester (Ctrl+R)
2. Choose:
 - EA name
 - Symbol and timeframe

- Date range
- “Every tick based on real ticks” for maximum accuracy

3. Check visual mode for signal tracking

4.2 Evaluating Performance

Important metrics:

- **Total Net Profit:** Overall profitability
- **Max Drawdown:** Risk exposure
- **Profit Factor:** Profitability vs loss ratio
- **Sharpe Ratio:** Risk-adjusted return
- **Number of Trades / Win Rate**

4.3 Optimization

Use MT5’s built-in optimizer:

- Test different EMA/RSI values
- Adjust SL/TP ranges
- Filter by:
 - Genetic algorithm
 - Walk-forward analysis

Avoid overfitting by validating with out-of-sample data.

5. Practical Risk Management Techniques

No strategy is complete without risk control:

5.1 Key Concepts

- **Risk per trade:** Usually 1-2%
- **Dynamic SL/TP:** Use ATR or volatility
- **Break-even logic:** Move SL after X pips
- **Trailing stop:** Lock in profits gradually
- **Trade filters:**
 - News filters (NFP, CPI, Fed)
 - Spread check before entry

5.2 Example: ATR-Based SL

```
double atr = iATR(NULL, 0, 14, 0);  
double sl = price - atr * 2;  
double tp = price + atr * 3;
```

6. Forward Testing and Going Live

Before live deployment:

- **Forward Test on Demo:** At least 2-4 weeks
- **Monitor Logs:** Ensure no runtime errors
- **Track Slippage & Spread**
- **Use a VPS:** 24/7 stability, especially for scalping EAs

Live Trade Checklist

- Broker conditions: ECN vs Market Maker
- Symbol suffix (e.g., XAUUSD.m)
- Correct GMT offset
- Set realistic expectations

7. Case Study: London Breakout EA on US30

Strategy Logic:

- Trade between 08:00-10:00 GMT
- If range breakout occurs (previous 1-hour high/low)
- Enter trade in breakout direction
- SL = 50 points, TP = 120 points
- Filter trades during high volatility only

This strategy, when coded and tested correctly in MQL5, can offer strong results with proper risk and time filtering.

Conclusion

The development of trading strategies in MQL5 is both a science and an art. It requires not only programming skill but a deep understanding of market dynamics. MetaTrader 5 offers a powerful ecosystem to build, test, and deploy automated strategies for retail traders and professionals alike.

By following a structured workflow—idea, rules, coding, testing, and optimization—you can build resilient and adaptive trading systems that stand the test of time and volatility.

MetaTrader 5 Trading Platform

MetaTrader 5 latest updates

News, implementations and technology

MetaTrader 5 User Manual

MQL5 language of trading strategies

MQL5 Cloud Network

End-to-End Analytics

Download MetaTrader 5

Install Platform

Uninstall Platform

About

Timeline

Terms and Conditions

Recurring Payment Agreement

Agency Agreement - Offer

Privacy and Data Protection Policy

Cookies Policy

Contacts and requests

MetaTrader 5

MQL5 Channels

Economic Calendar

Follow us on
socials for top
articles and
CodeBase updates

Not a broker, no real trading
accounts

35 Dodekanisou str, Germasogeia,
4043, Limassol, Cyprus

Copyright 2000-2025,
MetaQuotes Ltd