

Introduction

Our [Buddyllan](#) EA uses 4 main parameters that we wanted to optimize automatically each week to best match market variations.

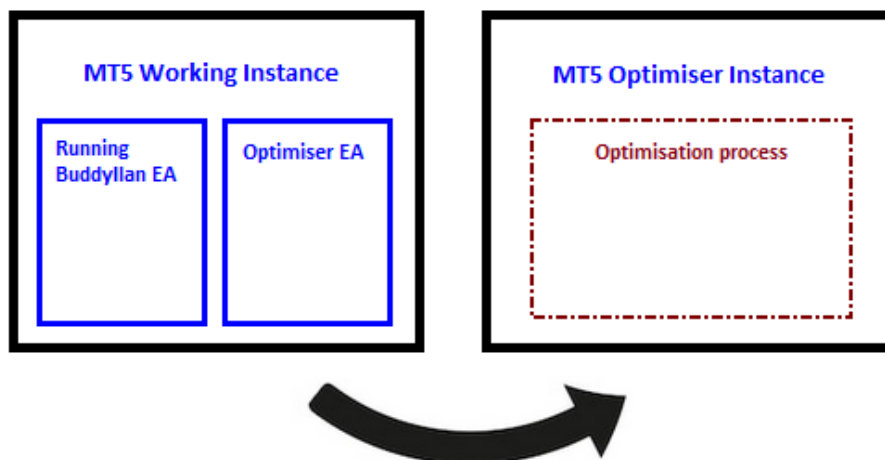
These parameters are :

- SL,
- TP,
- STOFILTER,
- STOTimeFrameFilter.

It is unrealistic to launch this type of process each week manually, so we looked for an existing mechanism to perform repetitive tasks but without success (for MT5) so we developed this one.

Thanks to Igor Malcev who write the article "[Automated Optimization of a Trading Robot in Real Trading](#)" for MT4.

Principe



A first MT5 instance is running 24/7, this instance hosts the Buddyllan EA and the EA on which we will work today (Optimiser EA) and which will launch on the second MT5 instance the optimization processes.

At the end of the process, the Optimiser EA will set optimised values in Global variables which will be read by the running Buddyllan EA.

The optimization is scheduled every Saturday without any manual intervention.

Copying data

As we said above, we need 2 MT5 instances.

The first MT5 instance is responsible for copying configuration, parameters and reports files between the 2 instances.

For security reasons, access to files outside a sandbox is not possible under MT5, so we will use the DOS command "xcopy" to copy the data between the 2 environments.

To do this we will need to use a Windows based DLL that we will declare as follows:

```
#import "shell32.dll"
int ShellExecutew(int hwnd,string Operation,string
                  File,string Parameters,string Directory,int ShowCmd);
#import
```

The call to this function will be done as follows:

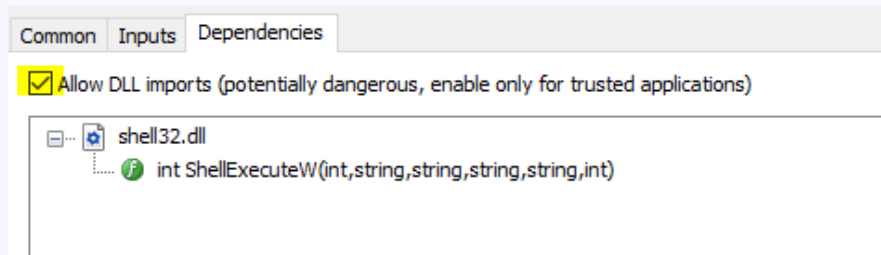
```
string PathIniFile = sTerminalTesterDataPath + "\\config\\common.ini";
string PathTester = TerminalInfoString(TERMINAL_DATA_PATH) + "\\MQL5\\Files\\Optimiser\\";
```

```
int ret = ShellExecuteW(0, "Open", "xcopy", "\"" + PathIniFile + "\" \"\" + PathTester + "\" /y", "", 0);
```

This function will also be called to start optimisation processes, e.g. :

```
int start = ShellExecuteW(0, "Open", sTerminalTesterPath + "\\terminal64.exe", "/config:" + Termi
if(start < 32)
{
    Print("Failed starting Tester");
    return false;
}
```

DLL import must be authorized for this EA :



Automated Optimisation

MT5 can be launched via online commands (see: [How to Start the Trading Platform](#)), automatic tasks can also be launched in this way.

For example, you can add a stanza "[Tester]" in the default configuration file (common.ini) to start an automatic optimisation at MT5 startup.

That's what we're gonna do.

Implementation

The Optimiser EA needs to know the paths of the MT5 Tester instance, they will be entered as parameters.

```
input string sTerminalTesterPath = "C:\\Program Files\\ForexTime MT5";
input string sTerminalTesterDataPath = "C:\\Users\\BPA\\AppData\\Roaming\\MetaQuotes\\Terminal\\5405
```

We have defined a working directory : "MQL5\\Files\\Optimiser" in the first MT5 instance.

Below the fonction "CopyAndMoveCommonIni()" copy the default configuration file "common.ini" of the MT5 Tester instance in our working directory and rename it as "optimise.ini".

```
bool CopyAndMoveCommonIni()
{
    string PathIniFile = sTerminalTesterDataPath + "\\config\\common.ini";
    string PathTester = TerminalInfoString(TERMINAL_DATA_PATH) + "\\MQL5\\Files\\Optimiser\\";

    int ret = ShellExecuteW(0, "Open", "xcopy", "\"" + PathIniFile + "\" \"\" + PathTester + "\" /y", "", 0);

    // wait until the file is copied
    Sleep(2500);
    if(ret < 32)
    {
        Print("Failed copying ini file");
        return false;
    }

    // We are working now in the sand box, we can use usual MT5 File commands
    string IniFileName = "Optimiser\\common.ini";
    string CopyTo = "Optimiser\\optimise.ini";
```

```

    return FileMove( IniFileName, 0, CopyTo, 0 );
}

```

For those interested, more information about the "ShellExecuteW" function can be found at this address: [ShellExecuteW](#). This function does not wait for the execution of the DOS command to return, hence the delay (Sleep 2500).

We now add the stanza "Tester" in this file :

```

bool AddTesterStanza()
{
    int filehandle = FileOpen("Optimiser\\Optimise.ini", FILE_READ|FILE_WRITE|FILE_TXT);

    if(filehandle!=INVALID_HANDLE)
    {
        FileSeek( filehandle, 0, SEEK_END );

        FileWrite(filehandle, "[Tester]\n",
            "Expert=BuddyIlan\\BuddyIlan\n",
            "ExpertParameters=BuddyIlanTester.set\n",
            "Symbol="+Symbol+"\n",
            "Period=M15\n",
            "Login=\n",
            "Model=4\n",
            "ExecutionMode=0\n",
            "Optimization=2\n",
            "OptimizationCriterion=0\n",
            "FromDate="+TimeToString(TimeGMT()-InpTesterPeriod*86400, TIME_DATE)+"\n",
            "ToDate="+TimeToString(TimeGMT(), TIME_DATE)+"\n",
            "ForwardMode=0\n",
            "Report=MYL5\\Files\\Reports\\BuddyIlanReport\n",
            "ReplaceReport=1\n",
            "ShutdownTerminal=1\n",
            "Deposit=10000\n",
            "Currency=EURUSD\n",
            "Leverage=1:100\n",
            "UseLocal=1\n",
            "UseRemote=0\n",
            "UseCloud=0\n",
            "Visual=1\n" );

        FileClose(filehandle);
    }
    else
    {
        Print("FileOpen, error ", GetLastError());
        return false;
    }

    return true;
}

```

In this stanza, we define the Expert we want to optimise ("Buddyllan") - this EA must be present in the second environment - and the ExpertParameters file as "BuddyllanTester.set" (be careful not to use the same name as your EA), we set the period (FromDate -ToDate) and all the parameters needed for the optimisation.

We set "ShutdownTerminal=1" which means that the terminal will shutdown at the end of the optimisation.

The Report will be generated in the file "Files\\Reports\\BuddyllanReport" - ".xml" extension will be added by the platform.

If your running EAs are hosted on a Virtual Server with low CPU resources, you can use some remote agents (see "UseRemote") for the optimisation process.

Parameter file

Then we have to create the parameter file we defined above (BuddyllanTester.set) which includes the values of each parameter of the EA (Buddyllan) that we want to optimize.

The default values of those parameters are set by the user (defined as parameters):

```
input _TradingMode TradingMode = Dynamic;           // Fixed or Dynamic volume
input double InpIlanFixedVolume = 0.1;              // Fixed volume size (if set)

input int InpNCurrencies = 1;                       // Number of Buddy Ilan instances on this acco

input double LotExponent = 1.4;
input bool DynamicPips = true;
input int DefaultPips = 15;

input int Glubina = 24;                             // Number of last bars for calculation of vola
input int DEL = 3;

input int TakeProfit = 40.0;                         // Take Profit (Point)
input int Stoploss = 1000.0;                        // Stop Loss (Point)

input bool InpIlanTrailingStop = true;              // Enable Trailing Stop
input int InpIlanDistanceTS = 5;                   // Trailing Stop distance (Point)

input int MaxTrades = 10;
input int InpDeviation = 10;                        // Max allowed price deviation (Points)

input bool bSTOFilter = true;                       // Dynamic Trend Filter
input bool bSTOTimeFrameFilter = false;            // Dynamic TimeFrame Filter
input int InpMaxTf = 60;                           // Max TimeFrame
```

The function below accepts 8 arguments, the first 4 correspond to the parameters to be optimized (SL, TP, STOFilter and STOTimeFrameFilter), if true, a "Y" will be positioned at the end of the corresponding parameter line. The following 4 arguments correspond to the already optimized values that we want to take into account during the next optimization.

As its name indicates, this function also copies the parameter file in the ad hoc directory (MQL5\Profiles\Tester) of the MT5 Tester instance.

```
bool CreateAndCopyParametersFile( bool SL, bool TP, bool STOFilter, bool STOTimeFrameFilter, int SLV
{
    int filehandle = FileOpen("Optimiser\\BuddyIlanTester.set",FILE_WRITE|FILE_TXT);

    if(filehandle!=INVALID_HANDLE)
    {
        FileWrite(filehandle,
            "_EA_IDENTIFIER=Buddy Ilan\n",
            "_EA_MAGIC_NUMBER=1111|0|1|10|\n",
            StringFormat("TradingMode=%d|0|0|0|\n", TradingMode ),
            StringFormat("InpIlanFixedVolume=%lf|0.0|0.000000|0.000000|\n", InpIlanFix
            StringFormat("InpNCurrencies=%d|0|1|10|\n", InpNCurrencies ),
            StringFormat("LotExponent=%lf|0.0|0.000000|0.000000|\n", LotExponent ),
            StringFormat("DynamicPips=%s|false|0|true|\n", (DynamicPips==true)?"true":
            StringFormat("DefaultPips=%d|0|1|10|\n", DefaultPips ),
            StringFormat("Glubina=%d|0|1|10|\n", Glubina ),
            StringFormat("DEL=%d|0|1|10|\n", DEL ),

            StringFormat("TakeProfit=%d|30|10|70|%\n", (TPValue == 0)?30:TPValue, (TP=
            StringFormat("Stoploss=%d|500|250|1500|%\n", (SLValue == 0)?1000:SLValue,

            StringFormat("InpIlanTrailingStop=%s|false|0|true|\n", (InpIlanTrailingSto
            StringFormat("InpIlanDistanceTS=%d|0|1|10|\n", InpIlanDistanceTS ),
            StringFormat("MaxTrades=%d|0|1|10|\n", MaxTrades ),
            StringFormat("InpDeviation=%d|0|1|10|\n", InpDeviation ),

            StringFormat("bSTOFilter=%s|false|0|true|%\n", (STOFilterValue==true)?"tru
            StringFormat("bSTOTimeFrameFilter=%s|false|0|true|%\n", (STOTimeFrameFilte
            StringFormat("InpMaxTf=%d|0|1|10|\n", InpMaxTf ) );

        FileClose(filehandle);
    }
    else
```

```

{
    Print("FileOpen BuddyIlanTester.set, error ", GetLastError());
    return false;
}

Sleep( 1500 );

string PathTester = TerminalInfoString(TERMINAL_DATA_PATH) + "\\MQL5\\Files\\Optimiser\\BuddyIlan
string PathProfile = sTerminalTesterDataPath + "\\MQL5\\Profiles\\Tester\\";

// copy the ini file into the tester folder
int ret = ShellExecuteW(0, "Open", "xcopy", "\"" + PathTester + "\" \"\" + PathProfile + "\" /y", "", 0);

// wait until the file is copied
Sleep(2500);
if(ret < 32)
{
    Print("Failed copying parameters file");
    return false;
}

return true;
}

```

Starting the Optimisation

The function below launches the Tester instance MT5, the optimization will be automatically launched using the parameters we have specified. This second instance will generate the result file and then it will shut down.

```

bool StartOptimizer()
{
    // Delete previous Report
    FileDelete( "Optimiser\\BuddyIlanReport.xml" );

    // Delete previous Report (second MT5 instance)
    string PathReport = sTerminalTesterDataPath + "\\MQL5\\Files\\Reports\\BuddyIlanReport.xml";

    ShellExecuteW(0, "Open", "cmd.exe", " /C del " + PathReport, "", 0);

    Sleep( 2500 );

    string sTerminalPath = TerminalInfoString(TERMINAL_PATH);

    // Start Optimisation process
    int start = ShellExecuteW(0, "Open", sTerminalTesterPath + "\\terminal64.exe", "/config:" + Termi
    if(start < 32)
    {
        Print("Failed starting Tester");
        return false;
    }

    Sleep( 15000 );

    return true;
}

```

From the first MT5 instance, the easiest way to find out if optimization is complete is to check if the report file is present.

When the report file is generated, we copy it into our working directory.

```

bool CopyReport()
{
    int nTry = 0;

    /*

```

```

* Waiting and copy Report file
*/

while( nTry++ < 500 )          // Timeout : 2 heures
{
    string PathReport = sTerminalTesterDataPath + "\\MQL5\\Files\\Reports\\BuddyIlanReport.xml";
    string PathTarget = TerminalInfoString( TERMINAL_DATA_PATH ) + "\\MQL5\\Files\\Optimiser\\";

    int ret = ShellExecuteW(0, "Open", "xcopy", "\"" + PathReport + "\" \"\" + PathTarget + "\" /y", "", 0);

    if( ret < 32 )
    {
        PrintFormat( "Waiting generation report (%d) ...", nTry );
        Sleep(15000);
    }
    else
    {
        if( FileExists( "Optimiser\\BuddyIlanReport.xml" ) == true )
        {
            PrintFormat( "Report found (ret=%d) ...", ret );
            Sleep(2500);
            return true;
        }
        else
        {
            PrintFormat( "Waiting report (%d) ...", nTry );
            Sleep(15000);
        }
    }
}

return false;
}

```

Reading the results

Report file is in XML format, fortunately, Paul van Hemmen wrote a library for MT5 to access this type of data, this library is available at this address : <https://www.mql5.com/en/code/1998> - many thanks to him.

We add this library in our EA as follows:

```
#include <EasyXML\EasyXml.mqh>
```

In fact, we added the function below and modified a few little things in this library to adapt it to our report files (see attached files).

```

//+-----+
//| Load XML by given file |
//+-----+
bool CEasyXml::loadXmlFromFullPathFile(string pFilename)
{
    string sStream;
    int iStringSize;

    Print("Loading XML File ", pFilename );
    int hFile=FileOpen(pFilename, FILE_ANSI|FILE_READ, 0, CP_UTF8);
    if(hFile==INVALID_HANDLE)
    {
        Err = EASYXML_ERR_CONNECTION_FILEOPEN;
        PrintFormat( "[%s] Err=%d", pFilename, GetLastError() );
        return(Error());
    }

    while(!FileIsEnding(hFile))
    {
        iStringSize = FileReadInteger(hFile, INT_VALUE);
        sStream += FileReadString(hFile, iStringSize);
    }
}

```

```

FileClose(hFile);

return(loadXmlFromString(sStream));
}

```

Access to the data is quite simple, several functions allow us to parse the results and read the data that interests us.

```

bool LoadResults( OptimisationType eType )
{
    /*
     * Init variable
     */

    BetterProfit = 0.0;

    /*
     * Load Results
     */

    CEasyXml EasyXmlDocument;
    EasyXmlDocument.setDebugging(false);

    if(EasyXmlDocument.loadXmlFromFullPathFile("Optimiser\\BuddyIlanReport.xml") == true )
    {
        str = "";

        CEasyXmlNode *RootNode = EasyXmlDocument.getDocumentRoot();

        for(int j=0; j<RootNode.Children().Total(); j++)
        {
            CEasyXmlNode *ChildNode=RootNode.Children().At(j);

            for(int i=0; i<ChildNode.Children().Total(); i++)
            {
                CEasyXmlNode *cNode=ChildNode.Children().At(i);

                if( cNode.getName() == "Worksheet" )
                {
                    switch( eType )
                    {
                        case _SL :

                            DisplayNodesSL(cNode);

                            PrintFormat( "-> SL=%d (Profit=%.2lf)", BetterSL, BetterProfit );

                            break;

                        case _TP :

                            DisplayNodesTP(cNode);

                            PrintFormat( "-> TP=%d (Profit=%.2lf DD=%lf)", BetterTP, BetterProfit, BetterDD );

                            break;

                        case _STO :

                            DisplayNodesSTO(cNode);

                            PrintFormat( "-> STOFilter=%s STOTimeFrameFilter=%s (Profit=%.2lf)", (BetterSTOFilter), (BetterSTOTimeFrameFilter), BetterProfit );

                            break;

                    }
                }

                break;
            }
        }
    }
}

```

```

    }
    }
}
else
    PrintFormat( "Error found" );

return true;
}

```

Since we want to optimize several parameters, we will analyze the results in different ways, so we need a specific function for each optimization. (SL, TP and STO parameters). These functions are recursive.

Below, the one used to analyze the results of the SL optimization:

```

void DisplayNodesSL( CEasyXmlNode *Node )
{
    for(int i=0; i<Node.Children().Total(); i++)
    {
        CEasyXmlNode *ChildNode=Node.Children().At(i);

        if( ChildNode.Children().Total() == 0 )
        {
            str += ChildNode.getValue() + ",";
        }
        else
        {
            DisplayNodesSL( ChildNode );

            if( Node.getName() == "Table" && ChildNode.getName() == "Row" )
            {
                string res[];
                StringSplit(str,',',res);

                // Bypass columns titles
                if( StringCompare( res[0], "Pass", true ) != 0 )
                {
                    double profit = StringToDouble(res[2]);
                    int sl = (int) StringToInteger(res[10]);

                    PrintFormat( "[%s] Profit=%.2lf StopLoss=%d DD=%s", str, profit, sl, res[8] );

                    if( profit > BetterProfit || (profit == BetterProfit && sl < BetterSL) )
                    {
                        BetterProfit = profit;
                        BetterSL = sl;
                    }
                }
            }

            if( Node.getName() == "Table" )
                str="";
        }
    }
}

```

This function is called on each row and cell.

If a node doesn't have any child, that means that it contains data, we store these data in a string that we split at the end of the line.

```

if( ChildNode.Children().Total() == 0 )
{
    str += ChildNode.getValue() + ",";
}

```

So, values for each column are available in the array "res[]" and we select the results of our choice.

EA Body

We now have all the necessary bricks to optimize our 4 parameters, deduce the best possible parameter setting and set the value of the corresponding global variables which will be read by the running Buddyllan EA.

```
void OnTimer()
{
    MqlDateTime dt;

    datetime now = TimeLocal( dt );

    // On Saturday
    if( dt.day_of_week != 6 )
    {
        bOptimisationDone = false;
        return;
    }

    // At 6:00 am
    if( dt.hour < 6 )
        return;

    // Already done ?
    if( bOptimisationDone == true )
        return;

    // Remove previous "optimise.ini"
    FileDelete( "Optimiser\\Optimise.ini" );

    // Create the EA config file and copy it to \MQL5\Profiles\Test (Tester Instance)
    if( CreateAndCopyParametersFile( true, false, false, false, 0, 0, true, false ) == false )
        return;

    // Copy common.ini -> optimise.ini
    if( CopyAndMoveCommonIni() == false )
        return;

    // Add [Tester] stanza in optimise.ini - https://www.metatrader5.com/en/terminal/help/start\_advan
    if( AddTesterStanza() == false )
        return;

    Print( "=====\nOptimisation SL-1" );

    // Start first optimisation SL
    StartOptimizer();

    // Copying the report file to the working directory
    if( CopyReport() == false )
        return;

    // Analyse reports
    if( LoadResults( _SL ) == false )
        return;

    Print( "=====\nOptimisation STO" );

    // Create parameter file for STO optimization (the 2 parameters will be optimised at the same time)
    if( CreateAndCopyParametersFile( false, false, true, true, BetterSL, 0, true, false ) == false )
        return;

    // Start optimizer STO
    StartOptimizer();

    // Copying the report file to the working directory
    if( CopyReport() == false )
        return;

    if( LoadResults(_STO) == false )
        return;
```

```

Print( "=====\nOptimisation SL-2" );

// Create parameter file for second SL optimization (recalculation with new STO parameter values)
if( CreateAndCopyParametersFile( true, false, false, false, 0, 0, BetterSTOFilter, BetterSTOTimeF
    return;

// Start optimizer
StartOptimizer();

if( CopyReport() == false )
    return;

if( LoadResults(_SL)==false )
    return;

Print( "=====\nOptimisation TP" );

// Create parameter file for TP optimization
if( CreateAndCopyParametersFile( false, true, false, false, BetterSL, 0, BetterSTOFilter, BetterS
    return;

// Start optimizer
StartOptimizer();

if( CopyReport() == false )
    return;

if( LoadResults( _TP ) == false )
    return;

/*
 * Conclusion
 */

PrintFormat( "=====\nSL=%d TP=%d STOFilter=%s STOTimeFrameFilter=%s (Profit=%.2
    BetterSL, BetterTP, (BetterSTOFilter==true?"true":"false", (BetterSTOTimeFrameFilter==true)"

/*
 * Set Global variables - The running BuddyIlan EA will read and use these new values
 */

// If the Draw Down found is over 50%, the EA Stop trading
if( BetterDD > 50.0 && GlobalVariableSet(gVarStop,1.0) == false )
{
    PrintFormat( "Error setting Global Variable [%s]", gVarStop );
}

if( GlobalVariableSet(gVarSL,BetterSL) == false )
{
    PrintFormat( "Error setting Global Variable [%s]=%d", gVarSL, BetterSL );
}

if( GlobalVariableSet(gVarTP,BetterTP) == false )
{
    PrintFormat( "Error setting Global Variable [%s]=%d", gVarTP, BetterTP );
}

if( GlobalVariableSet(gVarSTOFilter,(BetterSTOFilter==true)?1.0:0.0) == false )
{
    PrintFormat( "Error setting Global Variable [%s]=%.11f", gVarSTOFilter, (BetterSTOFilter==true
}

if( GlobalVariableSet(gVarSTOTimeFrameFilter,(BetterSTOTimeFrameFilter==true)?1.0:0.0) == false )
{
    PrintFormat( "Error setting Global Variable [%s]=%.11f", gVarSTOTimeFrameFilter, (BetterSTOTim
}

```

```

    bOptimisationDone = true;
}

```

Global variable names are built in the OnInit() function:

```

int OnInit()
{
    /*
     * Global variables
     */

    gVarStop = "BuddyIlan." + _Symbol + ".Stop";
    gVarSL = "BuddyIlan." + _Symbol + ".SL";
    gVarTP = "BuddyIlan." + _Symbol + ".TP";
    gVarSTOFilter = "BuddyIlan." + _Symbol + ".STOFilter";
    gVarSTOTimeFrameFilter = "BuddyIlan." + _Symbol + ".STOTimeFrameFilter";
}

```

Below, the full optimisation process:

```

2018.07.07 13:20:15.978 BuddyIlanOptimizer (EURGBP,M15) TERMINAL_PATH = C:\Program Files\MetaTrader
2018.07.07 13:20:15.978 BuddyIlanOptimizer (EURGBP,M15) TERMINAL_DATA_PATH = C:\Users\BPA\AppData\Ro
2018.07.07 13:20:15.978 BuddyIlanOptimizer (EURGBP,M15) TERMINAL_COMMONDATA_PATH = C:\Users\BPA\AppData
2018.07.07 13:20:32.586 BuddyIlanOptimizer (EURGBP,M15) =====
2018.07.07 13:20:32.586 BuddyIlanOptimizer (EURGBP,M15) Optimisation SL-1
2018.07.07 13:20:50.439 BuddyIlanOptimizer (EURGBP,M15) Waiting report (1) ...
2018.07.07 13:21:05.699 BuddyIlanOptimizer (EURGBP,M15) Waiting report (2) ...
2018.07.07 13:21:20.859 BuddyIlanOptimizer (EURGBP,M15) Waiting report (3) ...
2018.07.07 13:21:35.952 BuddyIlanOptimizer (EURGBP,M15) Report found (ret=42) ...
2018.07.07 13:21:38.471 BuddyIlanOptimizer (EURGBP,M15) Loading XML File Optimiser\BuddyIlanReport.x
2018.07.07 13:21:38.486 BuddyIlanOptimizer (EURGBP,M15) [0,11032.2600,1032.2600,3.3406,1.7096,1.5083
2018.07.07 13:21:38.487 BuddyIlanOptimizer (EURGBP,M15) [2,11463.8000,1463.8000,4.7837,2.0386,0.8454
2018.07.07 13:21:38.487 BuddyIlanOptimizer (EURGBP,M15) [4,11444.1000,1444.1000,4.7348,2.0340,0.8340
2018.07.07 13:21:38.487 BuddyIlanOptimizer (EURGBP,M15) [1,11297.1900,1297.1900,4.2392,1.8414,0.8180
2018.07.07 13:21:38.487 BuddyIlanOptimizer (EURGBP,M15) [3,11514.0800,1514.0800,4.9158,2.3170,1.4576
2018.07.07 13:21:38.487 BuddyIlanOptimizer (EURGBP,M15) -> SL=1250 (Profit=1514.08)
2018.07.07 13:21:38.487 BuddyIlanOptimizer (EURGBP,M15) =====
2018.07.07 13:21:38.487 BuddyIlanOptimizer (EURGBP,M15) Optimisation STO
2018.07.07 13:22:02.660 BuddyIlanOptimizer (EURGBP,M15) Waiting report (1) ...
2018.07.07 13:22:17.768 BuddyIlanOptimizer (EURGBP,M15) Waiting report (2) ...
2018.07.07 13:22:32.856 BuddyIlanOptimizer (EURGBP,M15) Waiting report (3) ...
2018.07.07 13:22:47.918 BuddyIlanOptimizer (EURGBP,M15) Waiting report (4) ...
2018.07.07 13:23:02.982 BuddyIlanOptimizer (EURGBP,M15) Report found (ret=42) ...
2018.07.07 13:23:05.485 BuddyIlanOptimizer (EURGBP,M15) Loading XML File Optimiser\BuddyIlanReport.x
2018.07.07 13:23:05.499 BuddyIlanOptimizer (EURGBP,M15) [0,11463.5000,1463.5000,4.4483,2.0614,0.8452
2018.07.07 13:23:05.499 BuddyIlanOptimizer (EURGBP,M15) [1,11444.1000,1444.1000,4.7348,2.0340,0.8340
2018.07.07 13:23:05.499 BuddyIlanOptimizer (EURGBP,M15) [2,11430.5300,1430.5300,5.1090,2.1548,0.8917
2018.07.07 13:23:05.499 BuddyIlanOptimizer (EURGBP,M15) [3,11470.7100,1470.7100,6.2851,1.8978,0.8146
2018.07.07 13:23:05.499 BuddyIlanOptimizer (EURGBP,M15) -> STOFilter=true STOTimeFrameFilter=true (P
2018.07.07 13:23:05.500 BuddyIlanOptimizer (EURGBP,M15) =====
2018.07.07 13:23:05.500 BuddyIlanOptimizer (EURGBP,M15) Optimisation SL-2
2018.07.07 13:23:29.921 BuddyIlanOptimizer (EURGBP,M15) Waiting report (1) ...
2018.07.07 13:23:45.043 BuddyIlanOptimizer (EURGBP,M15) Waiting report (2) ...
2018.07.07 13:24:00.170 BuddyIlanOptimizer (EURGBP,M15) Waiting report (3) ...
2018.07.07 13:24:15.268 BuddyIlanOptimizer (EURGBP,M15) Waiting report (4) ...
2018.07.07 13:24:30.340 BuddyIlanOptimizer (EURGBP,M15) Report found (ret=42) ...
2018.07.07 13:24:32.854 BuddyIlanOptimizer (EURGBP,M15) Loading XML File Optimiser\BuddyIlanReport.x
2018.07.07 13:24:32.872 BuddyIlanOptimizer (EURGBP,M15) [0,9269.9000,-730.1000,-2.7760,0.7328,-0.364
2018.07.07 13:24:32.872 BuddyIlanOptimizer (EURGBP,M15) [4,11470.7100,1470.7100,6.2851,1.8978,0.8146
2018.07.07 13:24:32.872 BuddyIlanOptimizer (EURGBP,M15) [3,11475.9500,1475.9500,6.2806,1.8995,0.8175
2018.07.07 13:24:32.872 BuddyIlanOptimizer (EURGBP,M15) [2,11400.7500,1400.7500,5.8609,1.8442,0.7759
2018.07.07 13:24:32.872 BuddyIlanOptimizer (EURGBP,M15) [1,10662.5500,662.5500,2.8807,1.3618,0.3815,
2018.07.07 13:24:32.873 BuddyIlanOptimizer (EURGBP,M15) -> SL=1250 (Profit=1475.95)
2018.07.07 13:24:32.873 BuddyIlanOptimizer (EURGBP,M15) =====
2018.07.07 13:24:32.873 BuddyIlanOptimizer (EURGBP,M15) Optimisation TP
2018.07.07 13:24:57.175 BuddyIlanOptimizer (EURGBP,M15) Waiting report (1) ...
2018.07.07 13:25:12.311 BuddyIlanOptimizer (EURGBP,M15) Waiting report (2) ...
2018.07.07 13:25:27.491 BuddyIlanOptimizer (EURGBP,M15) Waiting report (3) ...

```

```

2018.07.07 13:25:42.613 BuddyIlanOptimizer (EURGBP,M15) Waiting report (4) ...
2018.07.07 13:25:57.690 BuddyIlanOptimizer (EURGBP,M15) Report found (ret=42) ...
2018.07.07 13:26:00.202 BuddyIlanOptimizer (EURGBP,M15) Loading XML File Optimiser\BuddyIlanReport.x
2018.07.07 13:26:00.219 BuddyIlanOptimizer (EURGBP,M15) [1,11768.5700,1768.5700,8.2259,2.4484,1.1024
2018.07.07 13:26:00.219 BuddyIlanOptimizer (EURGBP,M15) [4,12343.5200,2343.5200,13.5464,2.5709,1.334
2018.07.07 13:26:00.219 BuddyIlanOptimizer (EURGBP,M15) [0,11243.4600,1243.4600,5.2913,1.6399,0.6887
2018.07.07 13:26:00.219 BuddyIlanOptimizer (EURGBP,M15) [3,12292.3500,2292.3500,11.8162,2.5837,0.925
2018.07.07 13:26:00.219 BuddyIlanOptimizer (EURGBP,M15) [2,12146.3900,2146.3900,11.0639,2.4416,1.222
2018.07.07 13:26:00.219 BuddyIlanOptimizer (EURGBP,M15) -> TP=70 (Profit=2343.52 DD=15.038900)
2018.07.07 13:26:00.219 BuddyIlanOptimizer (EURGBP,M15) =====
2018.07.07 13:26:00.219 BuddyIlanOptimizer (EURGBP,M15) SL=1250 TP=70 STOFILTER=true STOTIMEFRAMEFIL
2018.07.07 13:26:00.219 BuddyIlanOptimizer (EURGBP,M15) =====

```

Conclusion

The implementation of this process requires a minimum knowledge of MT5, its optimization mechanisms and programming.

Attached are the sources of this EA and the modified file "EasyXml.mqh" (must be copied in "MQL5\Include\EasyXml").

Hope this helps.