Manual de Desarrollo de Software
FxDreema Expert Advisor Builder

Autores

JULIAN BARNEY JAIMES RINCON

NANCY TAVERA CASTILLO

COEDICION

Unidades Tecnológicas de Santander Grupo de Investigación GRIIS Facultad de Ingeniería de Sistemas

ISBN - 383405

DEDICATORIA

Este libro va dedicado a todas las personas interesadas en aprender sobre robots financieros, trading y aprender sobre lo relacionado con el mercado de divisas con operaciones autónomas para sacarle el máximo beneficio posible.

A la UTS por brindar la oportunidad para realizar este tipo de proyectos en los cuales se incentiva a la comunidad a aprender sobre trading y los componentes que este conlleva.

INDICE

Introducción

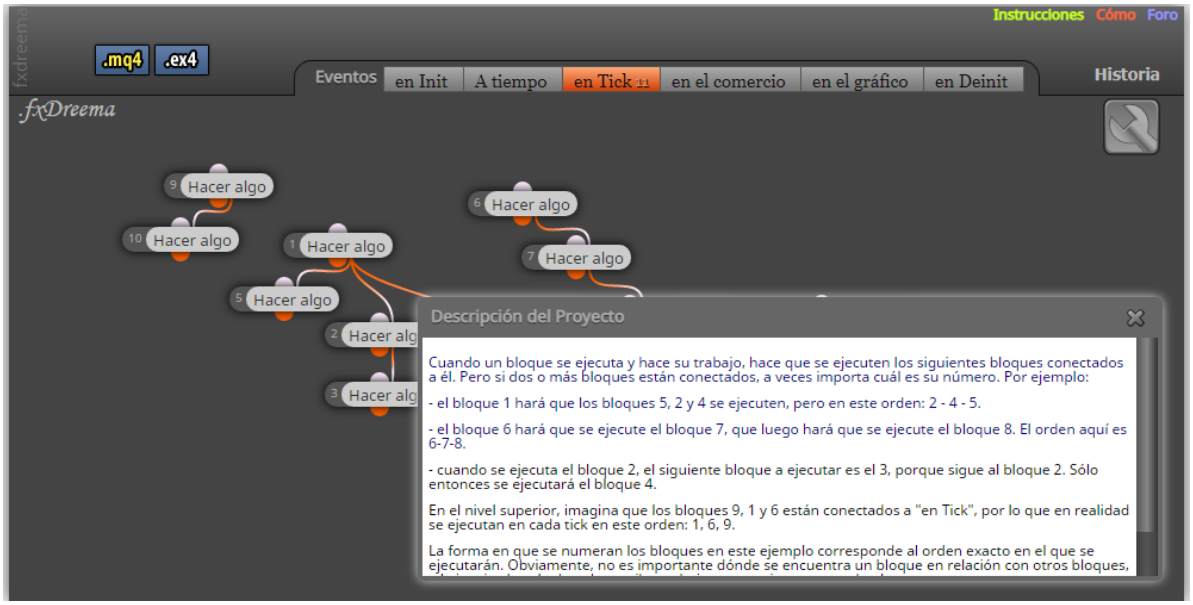
Es una herramienta de software para ayudarle a crear programas como Asesores Expertos y Scripts para la plataforma MetaTrader 4 y Metatrader 5 (MetaQuotes Software Corp).

La manera natural de crear estos programas es codificarlos manualmente usando lenguajes de programación especiales, como MQL4 o MQL5. Claro no todos son programadores, por lo que, si solo se necesita probar algunas estrategias comerciales sin codificarlas desde cero, se está en el lugar indicado.

Aún más, esta plataforma también se adecua a los programadores, por la facilidad de hacer cambios pequeños o grandes en el programa.

Funcionamiento

Fx Dreema permite conectar bloques de un árbol lógico. Cada uno de los bloques realiza una acción compleja y puede ajustarlo editando sus parámetros. El objetivo final es generar un programa, el producto de salida de FxDreema.



Cada bloque es un módulo, una caja negra diseñada para hacer alguna acción, ya sea para verificar ciertas condiciones para realizar alguna acción. Cada bloque tiene una entrada y al menos una salida. Cuando el bloque hace lo que debe hacer, activa su salida, lo que significa que, si hay otros bloques conectados a este, serán los siguientes en ejecutarse. Hay 3 tipos de salidas:

- **El naranja:** Se activa cuando el bloque logra hacer lo que está diseñado para hacer.
- **El Amarillo:** está normalmente activo cuando el naranja no lo está, pero hay algunas excepciones.
- **El Gris:** en los bloques de negación (acción) y se activa cuando se produce un error dentro del bloque.

Al seleccionar los bloques adecuados y conectar sus salidas con las entradas de otros bloques, puede crear muchas estructuras comerciantes.

Cuando haya terminado, se puede exportar / guardar el programa real y usarlo en su terminal comercial. ¡Se recomienda encarecidamente realizar una prueba retrospectiva primero!

Funciones de fxDreema

Características generales

-Fx Dreema está en línea desde noviembre de 2011

-Fx Dreema usa bloques gráficos para construir una estrategia. Cada bloque tiene parámetros de entrada para cambiar su comportamiento.

-El producto de salida de Fx Dreema es un archivo con una de las siguientes extensiones: .mq4,. ex4, .mq5 o. ex5

-Compatible con MetaTrader 4 y MetaTrader 5

-No, no se puede utilizar para crear indicadores personalizados hasta el momento

-Se puede utilizar de manera gratuita. Deja de exportar archivos cuando se alcanza cierto número de conexiones entre bloques

-Es una aplicación web, pero también está disponible como programa de Windows y, en este caso, puede funcionar sin conexión.

-Soporte: En este momento ya hay varias preguntas que tienen respuesta. Puedes preguntar algo en el correo electrónico o en el Foro (preferiblemente, porque la información se queda ahí y otras personas pueden usarla)

Principales características del constructor

-Trabaje con eventos como: Init, Timer, Tick, Trade,Chart o Denit

-Trailing stop: también puede rastrear take-profit, órdenes pendientes o grupo de operaciones (en MT4)

-Se pueden utilizar constantes globales (entradas del proyecto y variables

-Cubrir los gastos

-Gestión de dinero: % de riesgo, Fibonacci, Martingala totalmente personalizable, Secuencia personalizada y más

-verifique los valores dinámicos fácilmente (a partir de indicadores, por ejemplo)

-Dibujar y controlar objetos gráficos en la ventana principal del gráfico o en una subventana: flechas, líneas, texto, Fibonacci y otros

-Envíe mensajes a correo electrónico, sitio web o teléfono inteligente

-Escribir datos en un archivo

-Variables terminales

-Los bloques personalizados se pueden crear y utilizar en proyectos

-Cortar, Copiar, Pegar

Historial de acciones, para que pueda deshacer algunas acciones en cualquier comento

No es necesario guardar un proyecto, solo marque el punto actual en el Historial de acciones y no se eliminará

-Una copia del proyecto se almacena en los archivos de código fuente que exporta

-Los archivos de código fuente (construidos con fx Dreema) se pueden volver a importar a fxDreema, que crea un nuevo proyecto

Especificaciones técnicas

-fxDreema funciona mejor en Chrome y también hay un complemento para descargas fáciles para Chrome. Firefox y Opera también están bien.

-Casi todo el código JavaScript para el navegador se escribe manualmente. Las conexiones entre bloques son elementos SVG (vector)

-Las cookies se utilizan para almacenar información de la sesión y localStorage se utiliza para almacenar scripts JavaScript y (todavía no) plantillas HTML

-El código de salida que crea el constructor es difícil de leer y no tan rápido, pero este es siempre el sacrificio cuando facilitamos la codificación

-nodeBB se usa para el foro y también para administrar cuentas de usuario

-fxDreema está alojado en un VPS por **Sheernox Technology Group (anteriormente Zap5 Networks)**

Preparar MetaTrader

Descargue e instale MetaTrader

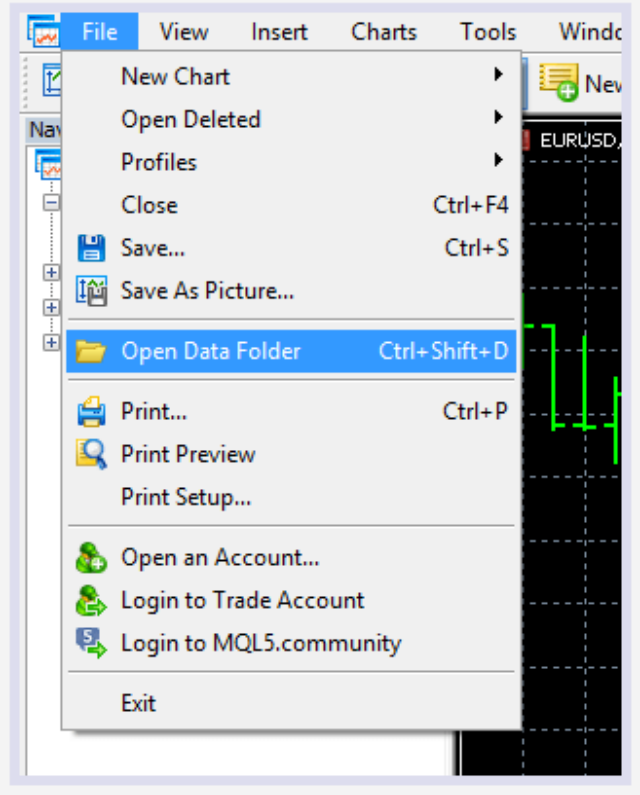
MetaTrader está desarrollado por MetaQuotes y probablemente ya lo tenga instalado en su PC. Si no lo tiene, debería poder descargarlo del sitio web de su corredor. Por si acaso, aquí es donde también puede encontrarlo:

-MetaTrader 4 (desarrollo detenido)

-MetaTrader 5

MetaTrader – Carpeta de datos

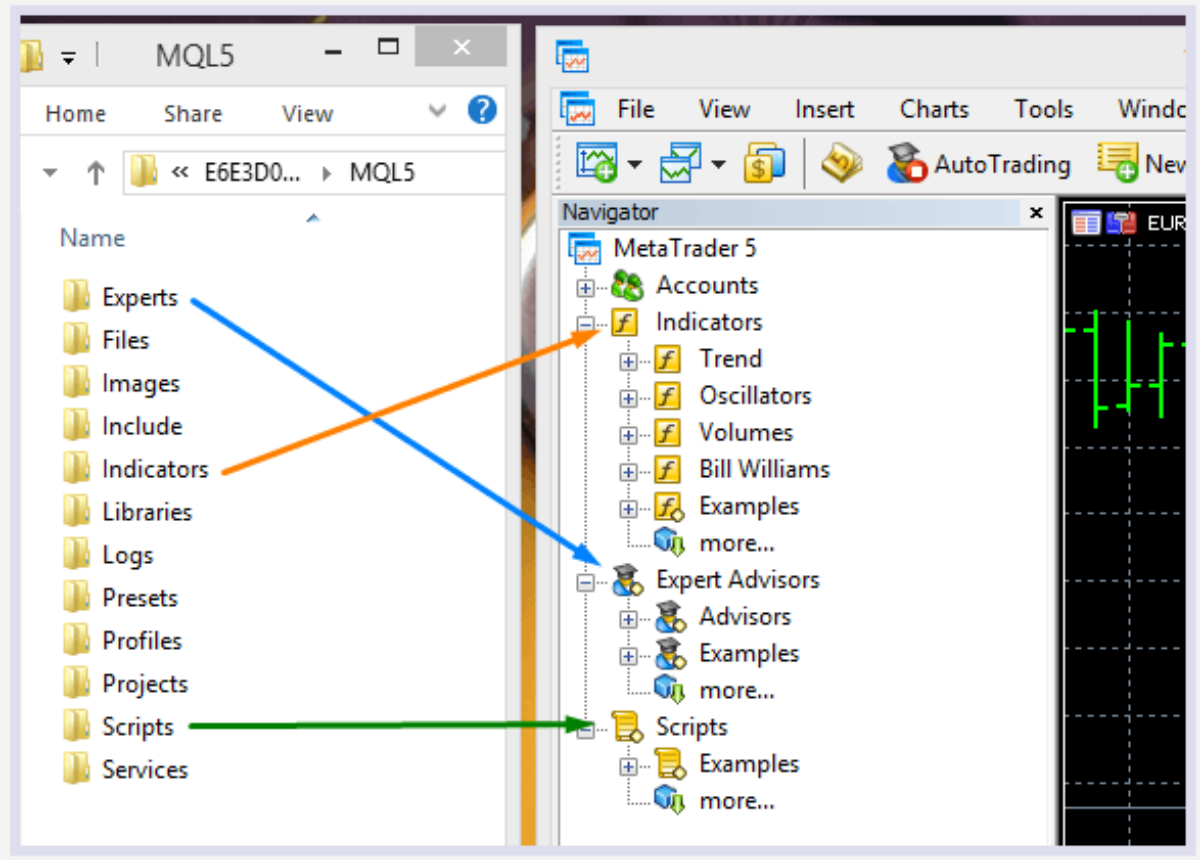
Debido a que el producto final de fxDreema es un robot comercial (Asesor Experto o un Script, que es un archivo, necesita saber dónde colocarlo. Sus robots comerciales, así como otros archivos de usuario, se encuentran en algún lugar de su disco duro. Para saber dónde, vaya a **Archivo -> Abrir carpeta de datos**



De forma predeterminada, su carpeta de datos está separada de la carpeta de instalación de MetaTrader por razones de seguridad y se encuantra en una ubicación similar a esta:

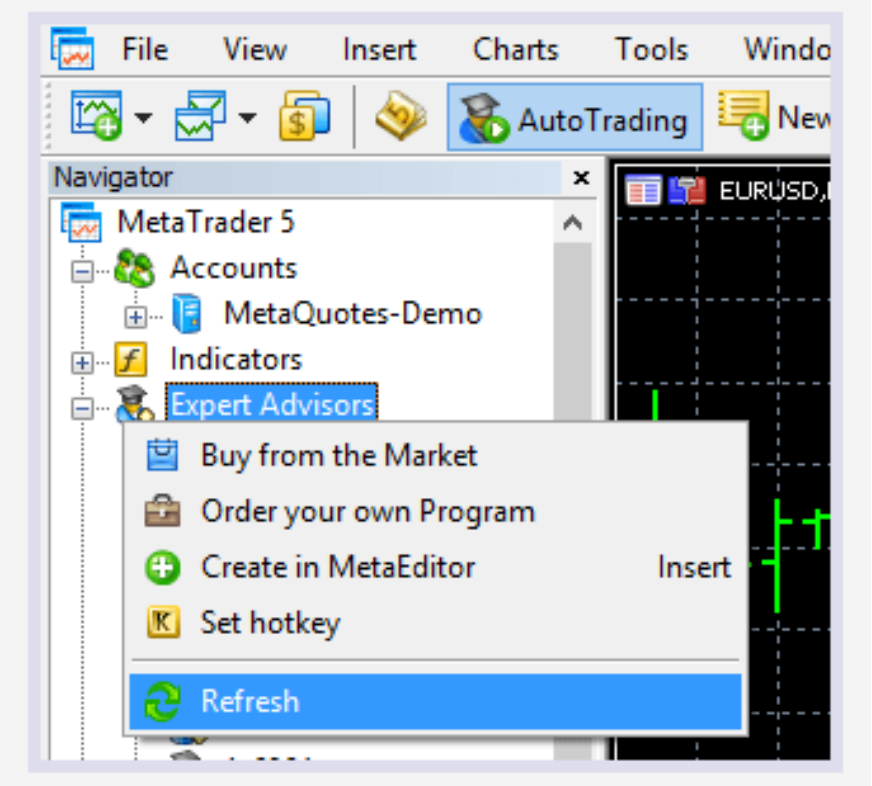
C: \ Users \ JohnSmith \ AppData \ Roaming \ MetaQuotes \ Terminal \ E6E3D0917DD641581E4779524EB3B1AA

Dentro de esta carpeta tiene otra carpeta -\ **MQL5** (o\ **MQL4,** dependiendo de la versión de **MetaTrader**) - en la que encontrará las carpetas *más importantes que utilizará:*



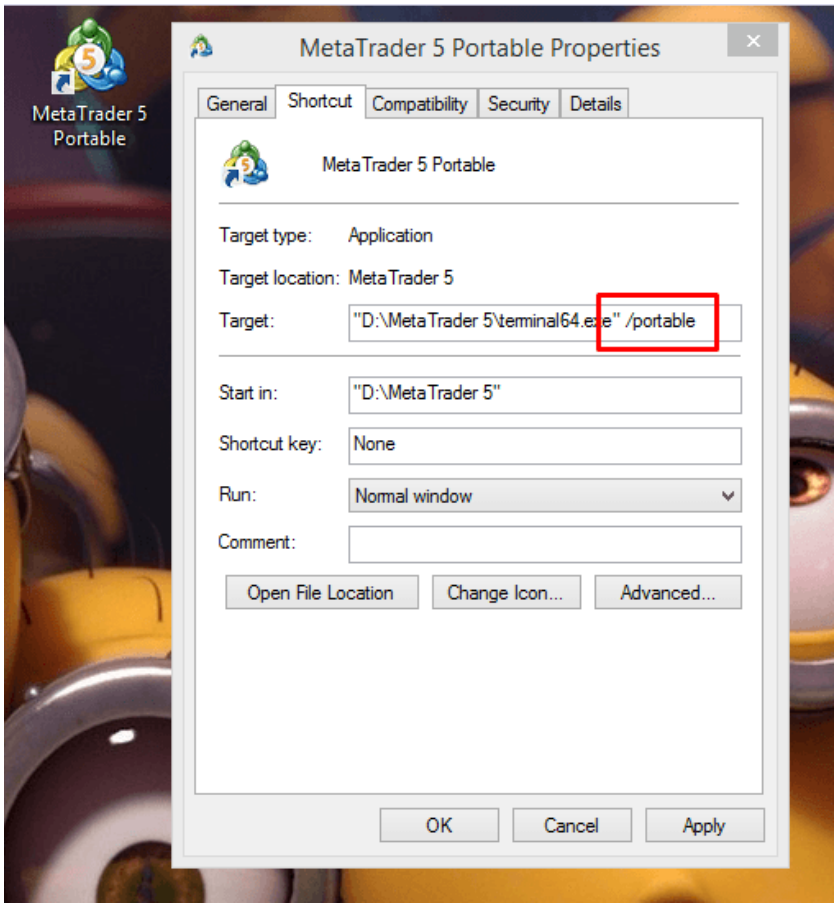
- **\ Expertos** es donde se encuentran sus asesores expertos
- **\ Indicators** es donde se encuentran sus indicadores personalizados
- **\ Scripts** es donde se encuentran sus scripts

Puede poner sus robots en estas carpetas. Cuando haga eso mientras **MetaTrader se** está ejecutando, también debe actualizar la lista en el **navegador**, como se muestra en la imagen a continuación. La otra forma es, por supuesto, reiniciar **MetaTrader**.



MetaTrader – Modo portátil

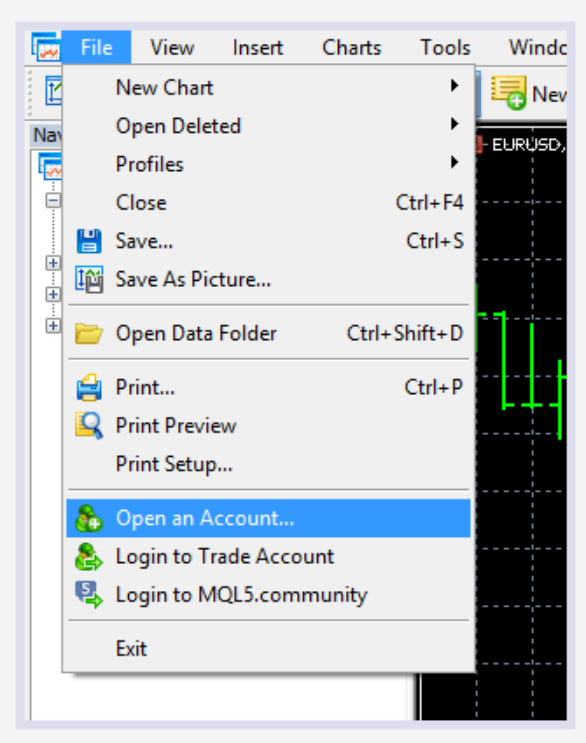
Si prefiere que su Metatrader esté fuera de \ **Archivos de programa** con todos los robots ubicados en su carpeta principal, puede forzar que se ejecute en modo portátil. Para hacer esto, edite las propiedades de su acceso directo como en la imagen de abajo. Agregue / **portable** (barra diagonal seguida de la palabra “portable”) al final de la propiedad Target no olvide cambiar el nombre de su atajo también.



Ahora, la opción **Abrir carpeta de datos** le enviará a **D**:\ **MetaTrader 5,** donde podrá encontrar su carpeta \ **MQL5** con todas las subcarpetas importantes.

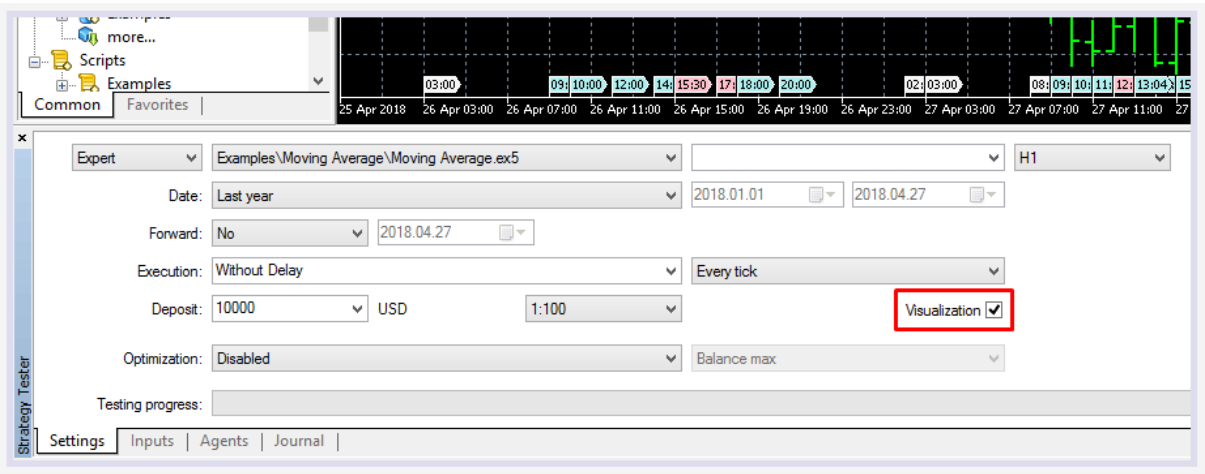
Crea una cuenta demo

Ahora puede crear una nueva cuenta demo en **Metatrader.** Es una buena idea probar siempre primero sus robots en cuenta Demo, porque no quiere perder dinero accidentalmente en su cuenta Real. Los errores acurren todo el tiempo mientras se desarrolla un programa.



Prueba en modo visual

En **MetaTrader 5** encontrará esta casilla de verificación llamada **Visualización.** En **MetaTrader 4,** la misma opción se llama **modo visual, es lo mismo.**



En este modo, puede realizar una prueba retrospectiva de su estrategia mientras la observa evolucionar en un gráfico separado. La velocidad es más rápida que en tiempo real, pero lo suficientemente lenta como para ver cualquier detalle de la Estrategia que ésta probando.

Descargar archivos fácilmente

Mientras trabaja en una estrategia, a menudo necesita verificar su progreso, por lo que a menudo necesita descargar archivos generados por **fxDreema.** En tu navegador normalmente usa el método **Guardar como**, pero esto rápidamente se vuelve abrumador. Por esta razón, existe una **extensión de Chrome** especial

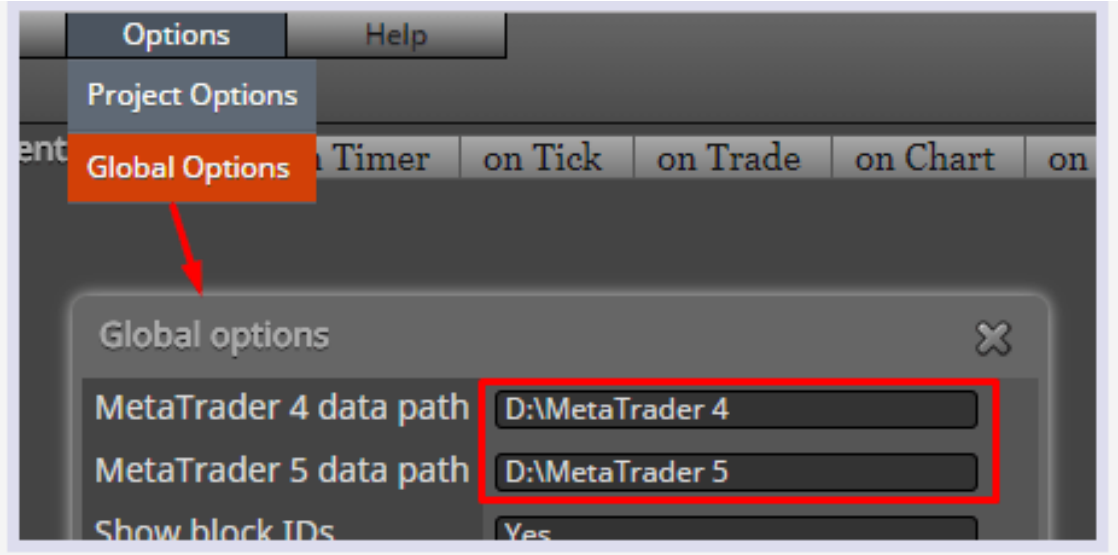
diseñada para descargar y escribir automáticamente los archivos de salida en la carpeta local adecuada.

-Obtenga **fxDreema: descargue archivos** de Chrome Web Store

Esta extensión usa la API de Chrome para hacer las cosas aburridas. Cuando descarga un archivo por primera vez, se utiliza el método **Guardar como.** Guardar ese archivo también otorga permiso a la extensión para descargar automáticamente cualquier archivo futuro con el mismo nombre de ahora en adelante- Por supuesto, la extensión también reconoce la página web de origen, no funciona fuera de **fxDreema.**

Si luego intenta descargar otro archivo con un nombre diferente, verá otra ventana **Guardar como.** Guarde ese archivo y no volverá a ver la ventana **Guardar como** para ese nombre de archivo. El mismo procedimiento ocurre para cada nombre de archivo único.

Luego, en fxDreema, debe configurar las rutas de datos, como se muestra en la imagen a continuación:



Las reglas básicas

Bloques y conexiones

Consulte esta rápida introducción al constructor.

Entonces, para hacer una estrategia de comercio aquí, debe conectar algunos bloques. Cada bloque está diseñado para algo específico. Suena simple, pero puede ser complicado, así que asegurarse de aprender cómo funciona todo.

Cuando un bloque se ejecuta, puede terminar su trabajo o no. Por ejemplo, hay un bloque de **Condición** que simplemente compara dos valores y dependiendo del resultado de esa comparación activa una de sus salidas. Las salidas son esos círculos en la parte inferior del bloque y puede usarlos para conectar los bloques

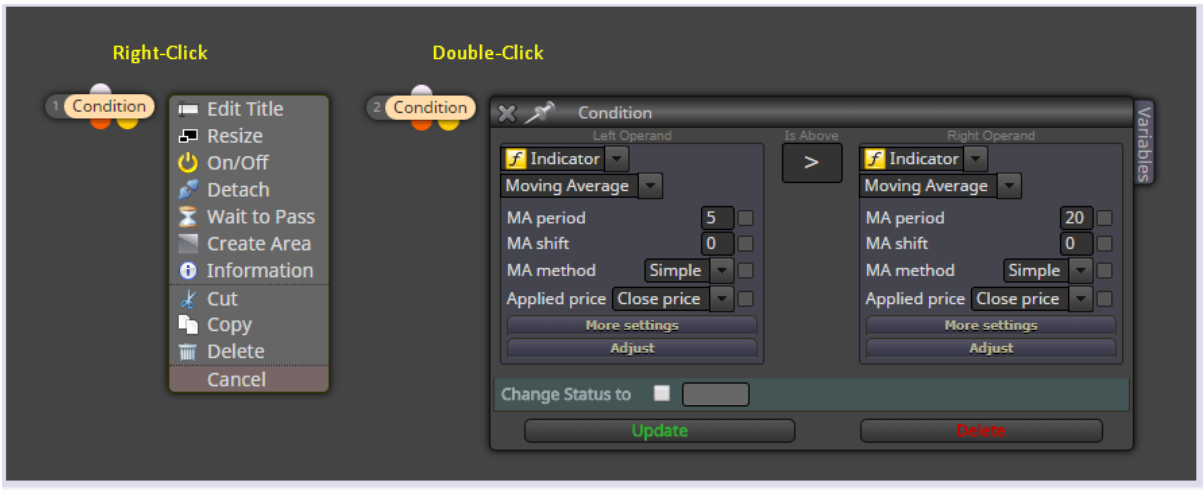
con otros bloques. Cuando una determinada salida se activa y hay otro bloque conectado a ella, ese bloque se activa.

Crear nuevos bloques

Simplemente **arrastre y suelte.** Arrastre un bloque de la lista y suéltelo en algún lugar sobre el archivo vacío

Puede hacer clic con el botón derecho y hacer doble clic

El **fxDreema Builder** es técnicamente una página web, pero actúa más como una aplicación de escritorio. En la mayoría de las páginas web, no tiene ninguna utilidad especial para hacer clic en el botón derecho, y el doble clic solo selecciona el contenido. Aquí tiene un menú contextual especial cuando hace clic en el bloque, por ejemplo, y puede abrir las propiedades del bloque cuando hace doble clic en él.



bloques desconectados no funciona

Si desea que cierto bloque haga algo, **debe** estar conectado a otro bloque. Los bloques desconectados se excluyen del archivo final generado. Puede tenerlos a mano en caso de que desee conectarse y usarlos en un futuro cercano, de lo contrario, son inútiles.

En caso de que desee ejecutar solo un bloque, conéctelo con el bloque **Pass. Pass** no hace nada por sí mismo, solo hace que el otro bloque se conecte, lo que significa que ingresa al archivo final generado.

Puede tener varias ramas de bloques

No es obligatorio comenzar con un solo bloque y conectar todo lo demás de ese bloque. Puede tener varias ramas de bloques y cada una de ellas funcionaría en su función. En la imagen de abajo, puede ver dos bloques o ramas: una que empieza con bloque #1 y otro que comienza con el bloque # 4. Debido a que ambas ramas están en el evento **on Tick,** ambas se ejecutarían en cada tick. Imagínese que el **de Tick** botón es un bloque y está conectado a los bloques # 1 y # 4.

Bloquear números casi no importa

Puedes ver que cada bloque tiene un número. Estos números no son tan importantes la mayor parte del tiempo, pero a veces pueden ser motivo de que alguna estrategia funcione de forma inesperada. Por ejemplo, la incertidumbre, el bloque con un número más bajo se ejecuta primero:

Los números de bloque pueden ser texto

El número de cualquier bloque se puede convertir en texto. Se recomienda usar solo caracteres latinos y números. No utilice caracteres especiales, ni siquiera espacios.

Copiar- Pegar bloques

Puede copiar y pegar bloques de la misma manera que copia y pega archivos, con clic derecho y Copiar y luego clic derecho y Pegar. Puede seleccionar varios bloques y copiarlos todos. También puede copiar bloques de un proyecto a otro.

No hagas esto demasiado en el mismo proyecto. Tener dos o más ramas de bloques con el mismo aspecto es una mala práctica. No siempre es posible, pero intente encontrar una forma de reutilizar la misma rama de bloques

Consejeros expertos

Que es un Asesor Experto

Un **Asesor Experto (EA)** es un pequeño programa que contiene un conjunto de instrucciones comerciales para MetaTrader. También puedes llamarlo **Robot,** porque hace algo en tu lugar. Los asesores expertos se pueden poner a trabajar en

tiempo real o se pueden realizar pruebas retrospectivas en períodos de tiempo anteriores.

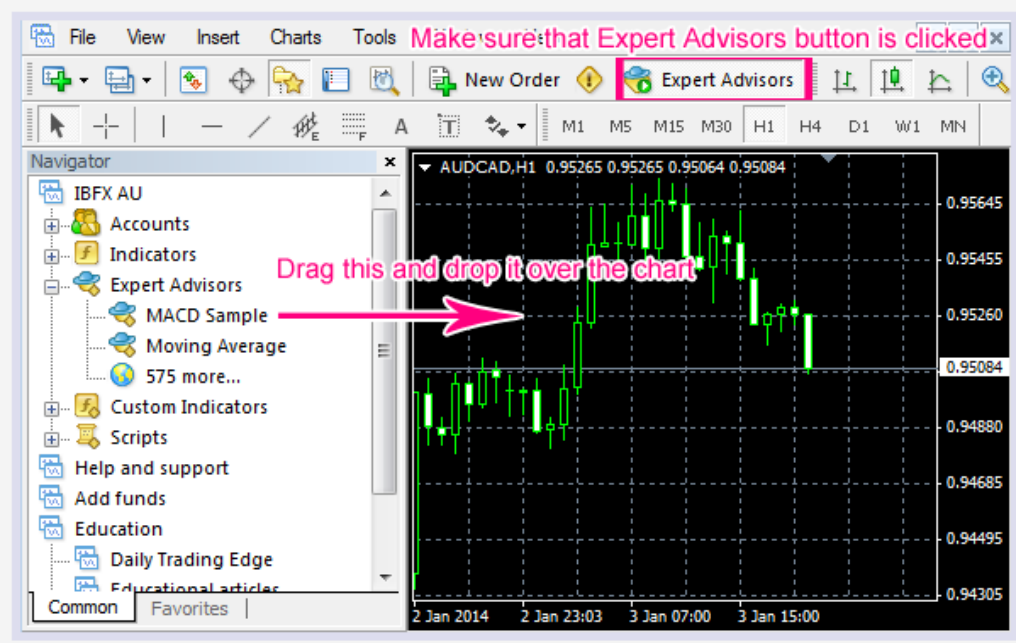
Los asesores expertos (EA) están escritos en el lenguaje de programación MQL4 (para MetaTrader 4) o MQL5 (para MetaTrader 5), y ninguno de ellos puede funcionar por sí solo como un programa independiente, para ejecutar cualquier EA, necesita MetaTrader.

MetaTrader carga su EA y lo alimenta con datos, controla sus eventos y ejecuta todas las instrucciones que contiene.

Los asesores expertos son archivos y puede encontrarlos en estas ubicaciones:

-Para MetaTrader 4: "**% Carpeta de datos %" / MQL4 / Experts /** "

-Para MetaTrader 5: "**% Carpeta de datos %" / MQL5 / Experts /** "



Puede ejecutar tantos EA como desee al mismo tiempo, pero solo puede tener 1 EA por gráfico.

Siempre prueba primero sus EA, luego pruébelos en una cuenta DEMO en vivo. Entonces, y solo si está seguro de que su EA funciona correctamente y no es peligroso, déjelo funcionar en una cuenta REAL.

Que puede hacer con un Asesor Experto

- Lea la información de su cuenta: saldo, capital, margen, apalancamiento, nombre del servidor, nombre de la cuenta y más
- Comunicarse con usted usando diferentes tipos de mensajes: mensaje de alerta, mensaje de impresión (pestañas de Expertos / Diario), cuadro de mensaje, escribir un comentario en el gráfico (esquina superior izquierda), reproducir sonido, enviar correo y más
- Lea los datos OHLC (open, High, Low, Close) de cualquier barra que pueda ver en el gráfico.

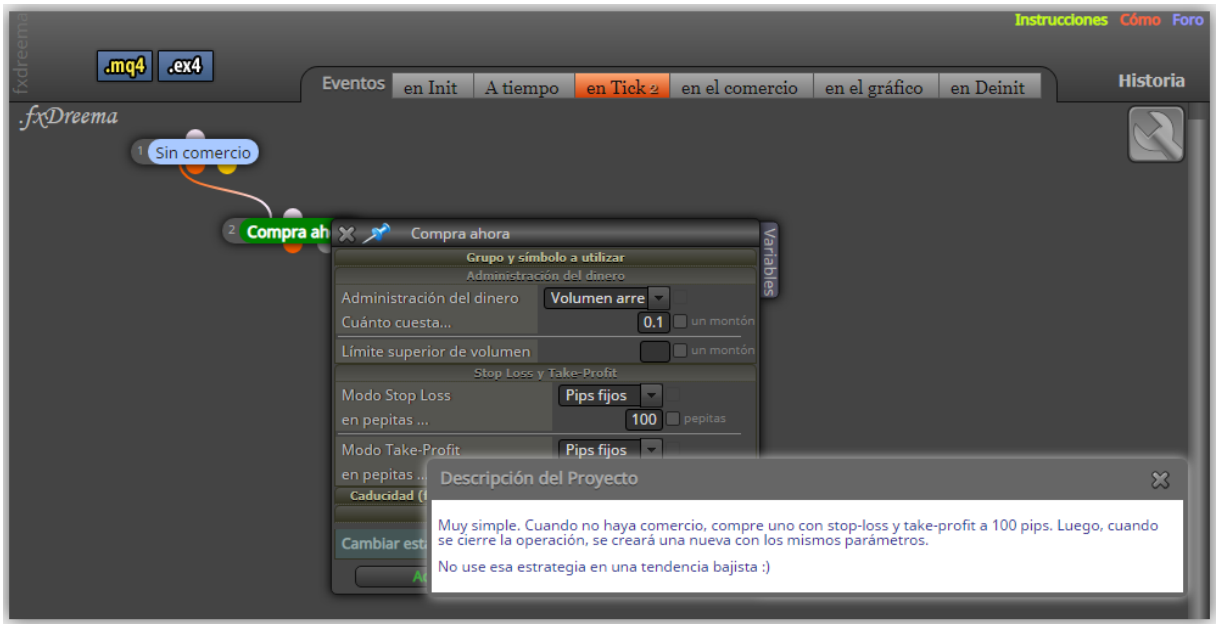
- Leer datos de indicadores para cada barra
- Leer archivos y escribir archivos
- Envíe / lea datos a otros EA utilizando variables globales (presione F3 en MetaTrader para ver la lista de variables globales)
- Crear diferentes objetos en el gráfico (flechas, líneas...) / Leer atributos de los objetos creados en el gráfico / Modificar sus atributos
- Crear operaciones y órdenes pendientes / Acceso a todas las operaciones y órdenes actuales y las del historial / Modificar sus parámetros (stop-loss, take-profit, tamaño de volumen)
- Haga cualquier cosa matemática y lógica como cualquier otro programa
- Y mucho más

Funcionamiento

En resumen, cada EA funciona así: **Evento => Verificaciones y cálculos => Acciones comerciales.**

En general, su EA debería obtener algunos datos de MetaTrader, realizar algunas verificaciones y cálculos con esos datos y luego decirle a MetaTrader qué acción comercial debe realizar: comprar, vender o lo que sea.

A continuación, puede ver un EA muy básico. Verá que los bloques se colocan bajo el evento **on Tick**, lo que significa que el bloque **No trade** se ejecuta en cada tick. Cuando no hay comercio, el bloque Comprar ahora se ejecuta y crea un nuevo comercio:



Para el ejemplo tenemos la siguiente estructura:

Evento (en Tick) **=> Verificaciones y cálculos** (sin comercio) => **Acciones comerciales** (Comprar ahora)

Eventos

Cuando **MetaTrader** detecta un **evento,** ejecuta cierta parte del código del EA. El evento más importante es cuando recibe un nuevo Tick y hay algunos otros eventos:

Es muy importante comprender que todos los EA funcionan debido a unos pocos **eventos** básicos. Lea sobre ellos a continuación.

-**Garrapata**: Se activa cuando aparece una nueva garrapata. Normalmente, la mayor parte de su EA funciona con cada tick.

-**Init**: Se activa una vez cuando se inicia el EA.

-**Deinit**: Se dispara una vez cuando se detiene el EA.

-**Comercio**: Se activa cada vez que abre, cierra o modifica operaciones u órdenes.

-**Temporizador:** Se dispara en cierto período de tiempo, por ejemplo, cada 60 segundos.

-**Gráfico:** Se activa cuando crea, modifica, elimina o hace clic en objetos en el gráfico.

Guiones

Definición

Los scripts son similares a los asesores expertos, pero hacen algo una vez. Pueden ejecutar un script haciendo doble clic en él en el navegador. Cuando el script hace todo su trabajo, lo que normalmente ocurre en menos de un segundo, se cierra automáticamente. Los scripts no funcionan en cada tick, simplemente hacen lo que se supone que deben hacer una vez, lo más rápido posible, y luego salen inmediatamente.

Los scripts son adecuados cuando se desea crear acciones comerciales simples, como cerrar todas sus operaciones con un solo clic (en realidad, doble). En cierto sentido, los scripts extienden la interfaz del terminal al agregarle acciones más rápidas.

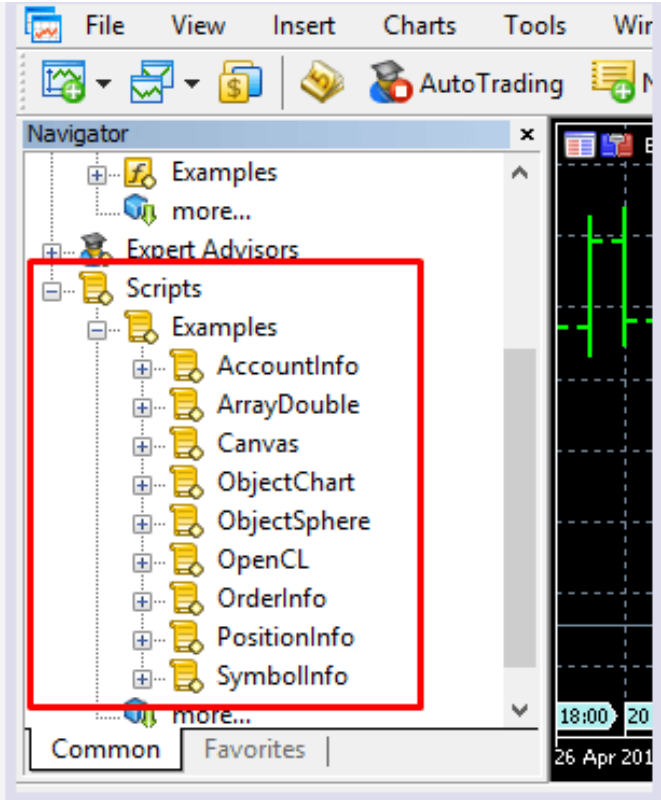
Aquí es donde se encuentran los scripts:

-Para MetaTrader 4:" **% Carpeta de datos % / MQL4 / Scripts /**"

-Para MetaTrader 5:" **% Carpeta de datos % / MQL5 / Scripts /**"

Coloque sus archivos .mq4 /. ex4 o mq5 /. ex5 en estas carpetas.

En MetaTrader, puede encontrar su script en **Scripts** en el panel del **navegador.**



Eventos

Solo hay un evento

-Inicio: se activa una vez cuando se inicia el script

Indicadores

Definición

El propósito de los indicadores es brindarle información visual que pueda ayudarlo mientras opera. Eta información puede verse como líneas, histogramas, flechas o cualquier otro tipo de información visual que puede ver con sus ojos y decidir cómo operar.

Los indicadores también pueden dar esa información a otros programas: asesores expertos o scripts. Pero en este caso la información no es visual, son solo **números,** porque los programas de computadora entienden los números.

Técnicamente, todos los indicadores funcionan de la misma manera: llenan algunas matrices con datos numéricos. Estas matrices se denominan **búferes de salida** (o simplemente **búferes**). **MetaTrader** lee esa información y la imprime en el gráfico.

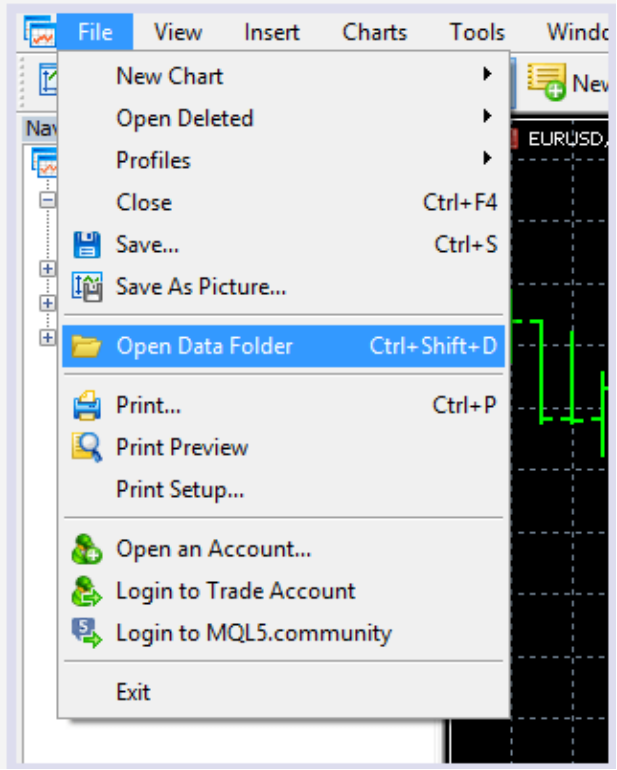
Encuentre sus indicadores personalizados

Existe un término **Indicadores personalizados:** Estos son los indicadores que puede escribir para **MetaTrader.** El propio **MetaTrader** tiene una lista de indicadores integrados que puede usar, pero no puede modificar. Puede encontrar sus indicadores personalizados en las siguientes carpetas:

-Para MetaTrader 4:” **% Carpeta de datos % / MQL4 / Indicadores /**”

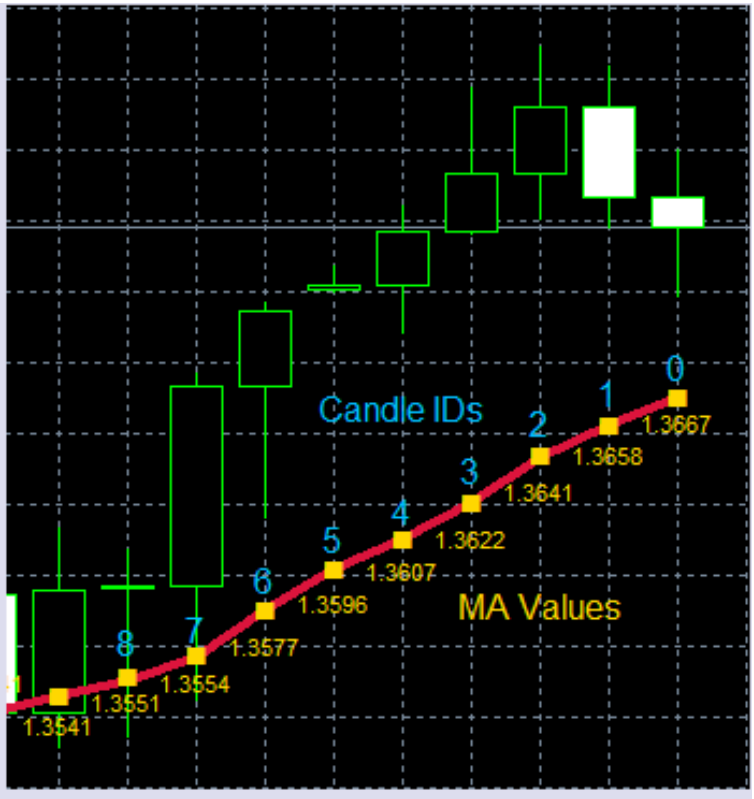
-Para MetaTrader 5:” **% Carpeta de datos % / MQL5 / Indicadores /**”

Así es como encuentra su carpeta de datos



Cómo se comunican los indicadores con otros programas

Ya sabemos que los indicadores colocan sus datos numéricos en algo llamado **búfer,** para que otros robots puedan leer esos datos. Un búfer es una matriz unidimensional llena de datos numéricos. El número total de elementos en esa matriz es el mismo que el número de velas en el gráfico, lo que significa que para cada vela hay un valor. Cada valor del búfer se asigna con una vela en particular. Tomemos el conocido promedio móvil, por ejemplo:



Cada vela tiene un número de identificación o turno. El número de la vela más nueva es 0. El número de la vela anterior es 1. Luego tenemos 2,3,4 y así sucesivamente. Para cada vela tenemos un valor numérico:

-El valor de la vela **0** es **1.3667**

-El valor de la vela **1** es **1.3658**

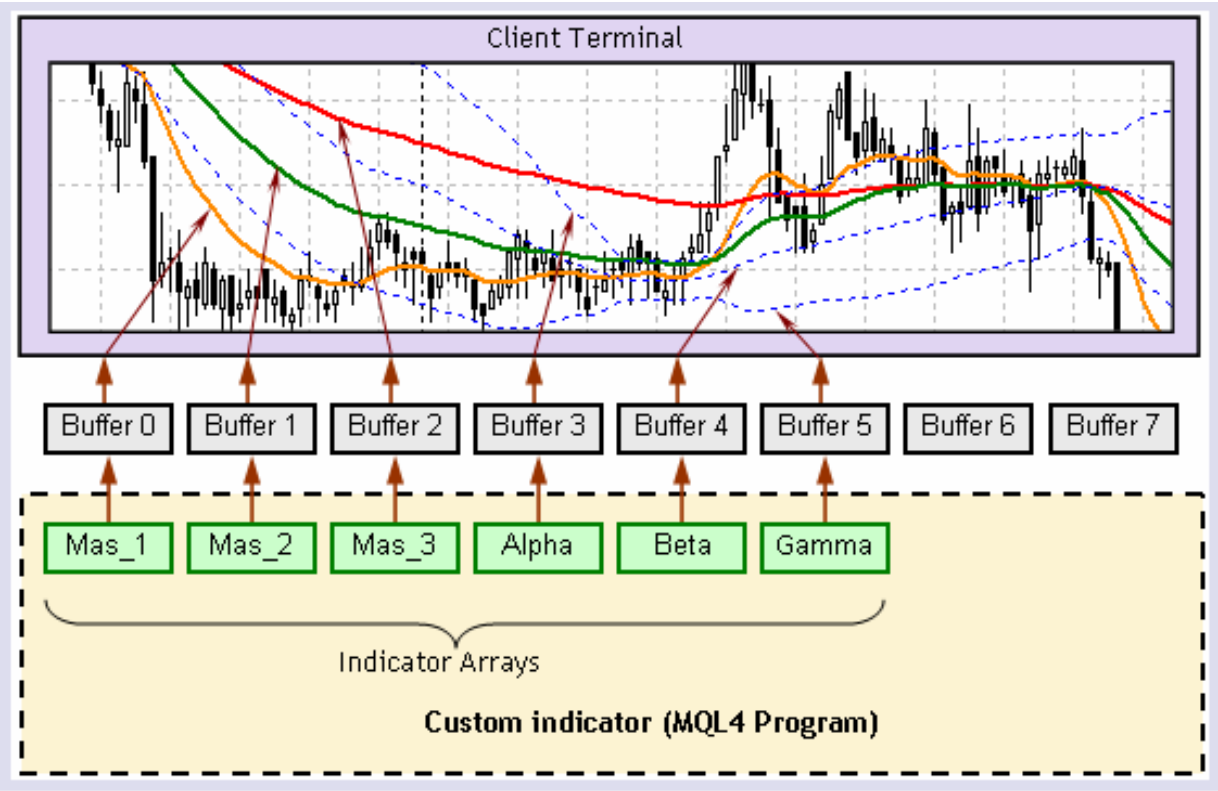
-El valor de la vela **2** es **1.3641**

Básicamente, esta es nuestra matriz de datos, el búfer de salida. Cuando Metatrader traza estos números en el gráfico, vemos una línea.

Podemos tener varios búferes y cada uno de ellos tiene su propio número. Similar a las velas, el primer búfer es 0, el segundo es 1, luego 2 y así sucesivamente.

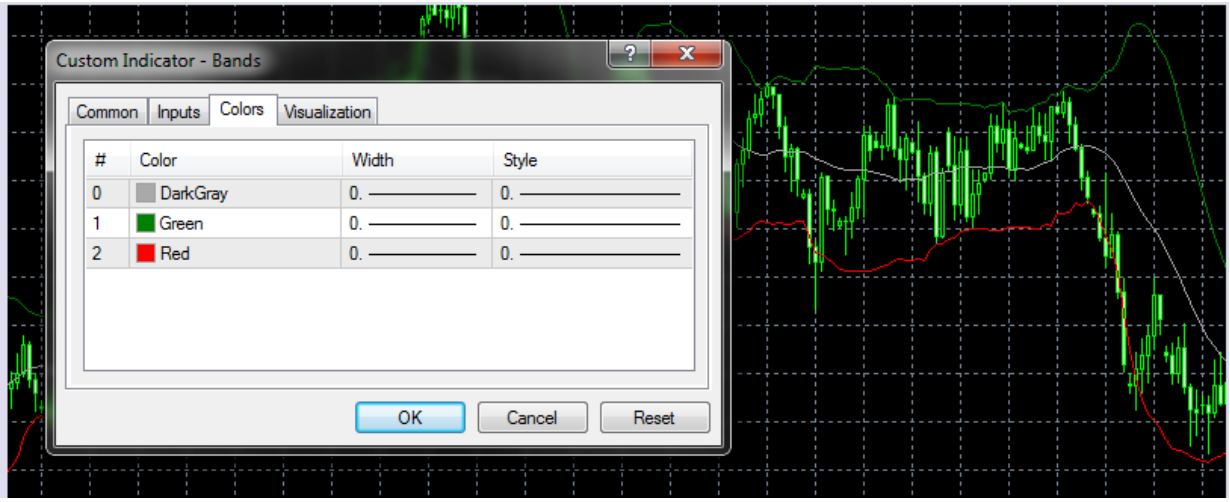
Ahora digamos que queremos obtener un valor de un indicador en nuestro Asesor Experto. Necesitamos apuntar al indicador (por nombre de archivo), al búfer de salida (0, 1, 2, 3...) y al ID de la vela (0, 1, 2, 3...).

Haga clic aquí para encontrar más información sobre los búferes de salida.



Como saber qué búferes tiene su indicador personalizado

Cuando agrega un indicador personalizado a un gráfico, puede ver sus parámetros. Vaya a la pestaña Colores y verá algo como esto:



Obviamente, hay 3 búferes de salida e incluso puedes ver sus números: 0,1 y 2. También puedes adivinar qué representa cada uno de ellos:

-El búfer **0** es para la **línea central**

-El búfer **1** es para la **línea superior**

-El búfer **2** es para la **línea inferior**

Indicadores de flechas (señales interrumpidas)

Algunos indicadores no imprimen líneas continuas. Probablemente haya visto indicadores con flechas colocadas aquí y allá. ¿Cómo lo hacen?

En el código del indicador hay instrucciones sobre cómo **MetaTrader** debe extraer los datos de los búferes en el gráfico. SI esos datos se verán como una línea, o como un histograma, o algún tipo de flechas o puntos. Estos son los posibles estilos de dibujo:

-Estilos de dibujo para MetaTrader 4

-Estilos de dibujo para MetaTrader 5

Pero ya sabemos que cada búfer de salida tiene tantos elementos (valores) como velas existen en el gráfico, y esto siempre es, hay un valor especial que podemos poner en el búfer y cuando **MetaTrader** ve ese valor, no muestra nada en el gráfico para esa vela. Este valor lo llamo **EMPTY_VALUE**. Esta es una constante predefinida y su valor real es el mayor valor entero posible:

-32 bits (MetaTrader 4): EMPTY_VALUE es igual a 2,147,483,647

-64 bits (MetaTrader 5): EMPTY_VALUE es igual a 9.223.372.038.854.775.807

Ahora puede imaginar que si el búfer de salida está lleno con EMPTY_VALUE hasta el final, no verá nada en el gráfico.

Sugerencia 1: En fxDreema, el bloque **Condición** no pararía si uno de sus operandos es igual a **EMPTY_VALUE.** Esto se debe a que **EMPTY_VALUE** debe tratarse como algo que no existe, su nombre contiene la palabra “vacío” después de todo.

Sugerencia 2: si espera señales de un indicador que dibuja flechas y no tiene ninguna, intente establecer el parámetro Candle ID en algo mayor que 0. Estos indicadores normalmente le dan señales de una vela más antigua, no en la actual. Pruebe con Candle ID=1.

Flechas y líneas en el gráfico, pero nada en los búferes

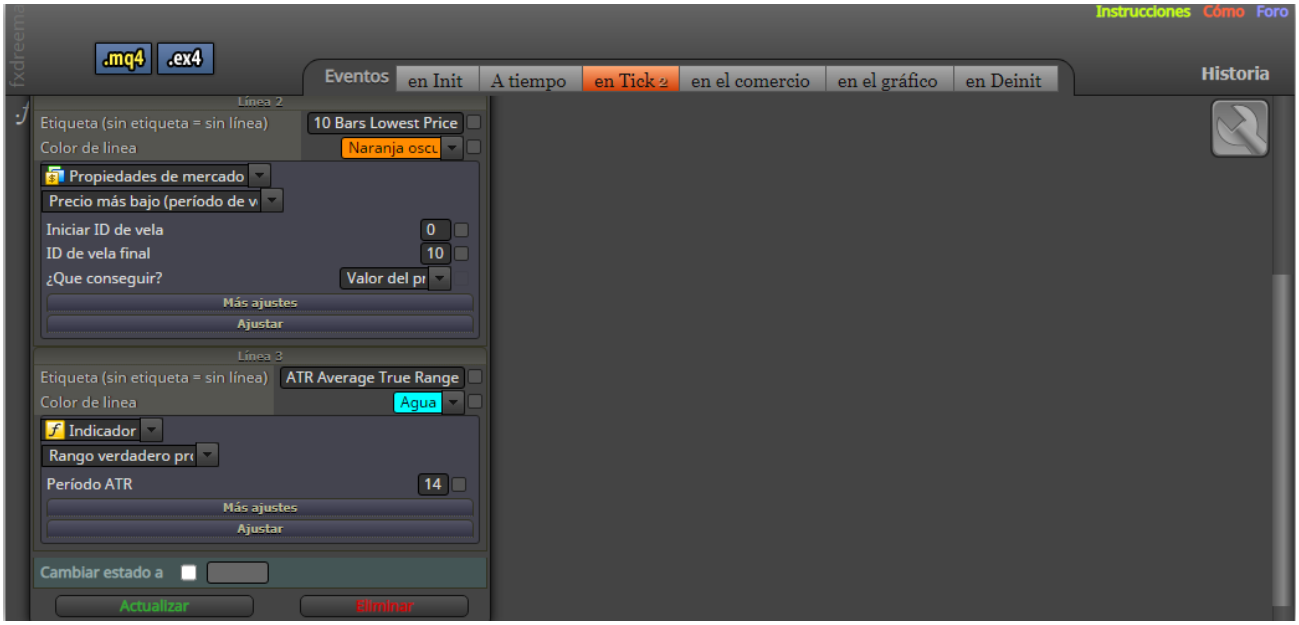
Si es posible. Algunos indicadores dibujan sus objetos directamente en el gráfico. Esto es cierto principalmente por los indicadores que trazan líneas de tendencia o flechas. En este caso, no se puede usar los búferes de salida para extraer los valores, pero puede obtener la información de los objetos. Aunque a menudo esta no es una tarea fácil.

También tenga en cuenta que, si este es el caso, si el indicador que está utilizando está dibujando solo objetos y en su **Asesor Experto** logró trabajar con ellos de alguna manera, la estrategia podría no funcionar cuando realiza **Optimización,** debido a algunas limitaciones en **MetaTrader**

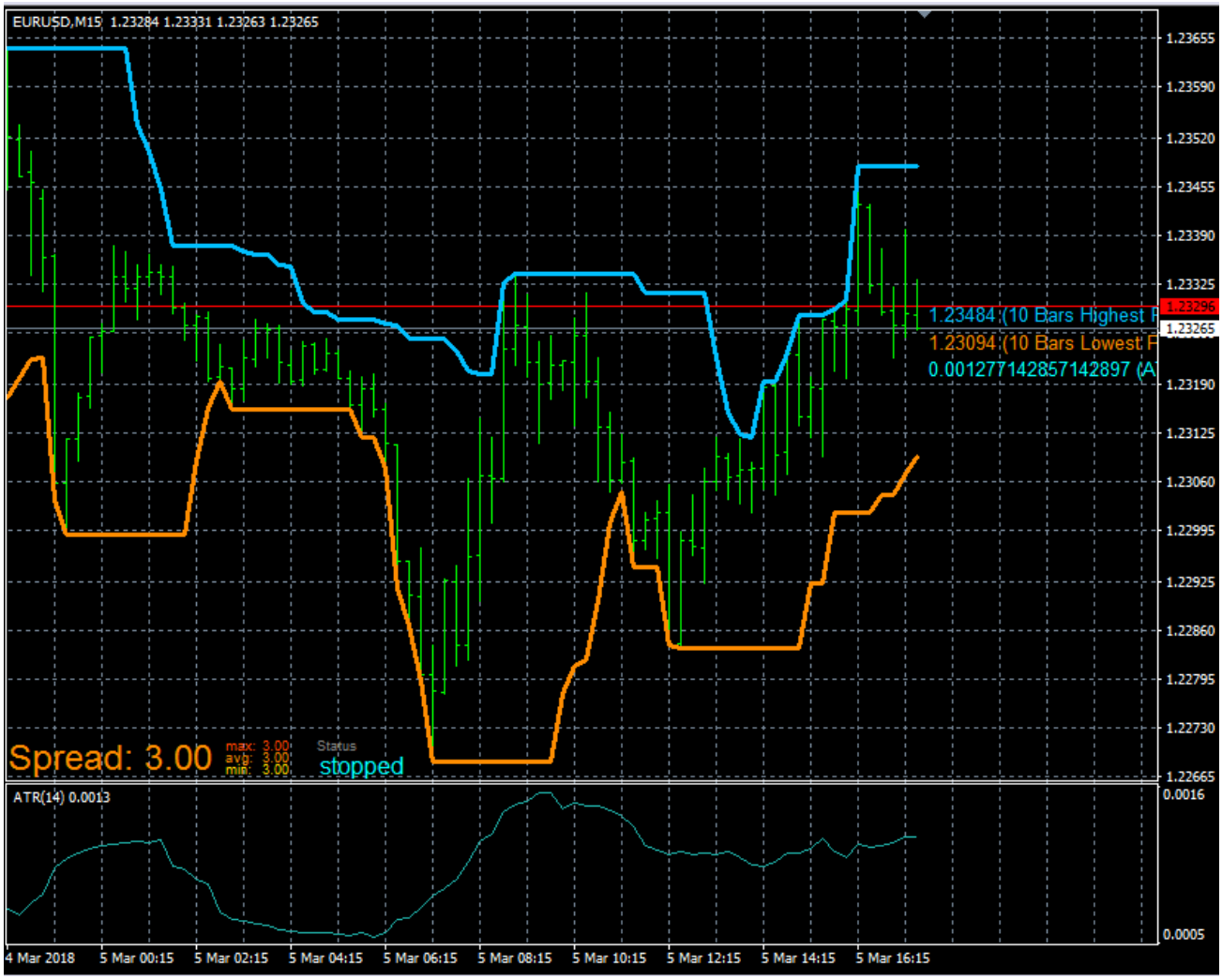
Haga clic aquí para obtener más información sobre los objetos de gráfico

Consulte el contenido del búfer

Cuando prueba por primera vez un nuevo indicador personalizado, probablemente no tenga ni idea de lo que hay dentro de sus búferes y cómo usarlos en sus estrategias. Pero puede verificarlos fácilmente con el bloque **Trace.** Eche un vistazo al siguiente ejemplo (Esto no es una captura de pantalla):



Cuando se realiza una prueba retrospectiva, el resultado se ve así:



Lo que quieres ver son varios valores. Si solo ve un valor que es el mismo para todas las velas, entonces no puede usar este búfer con esta configuración.

¿Cuándo se muestran los indicadores en el gráfico automáticamente?

Notará que todos los indicadores aparecen automáticamente después de una prueba de retroceso, pero no en una prueba de avance. ¿No te parece extraño?

Cuando un **Asesor Experto** o un **Script** está usando un indicador en su código, ese indicador se carga en segundo plano y nunca lo ven en el gráfico. Por eso, si desea ver el indicador, debe agregarlo manualmente en el gráfico. Pero cuando hace esto, en realidad está creando otra instancia del indicador y todavía no ve la que está funcionando en segundo plano. Incluso puede intentar modificar sus parámetros de entrada, esto no cambiará los parámetros del indicador que es cargado por el robot en segundo plano.

Para Obtener más información detallada, lea este articulo

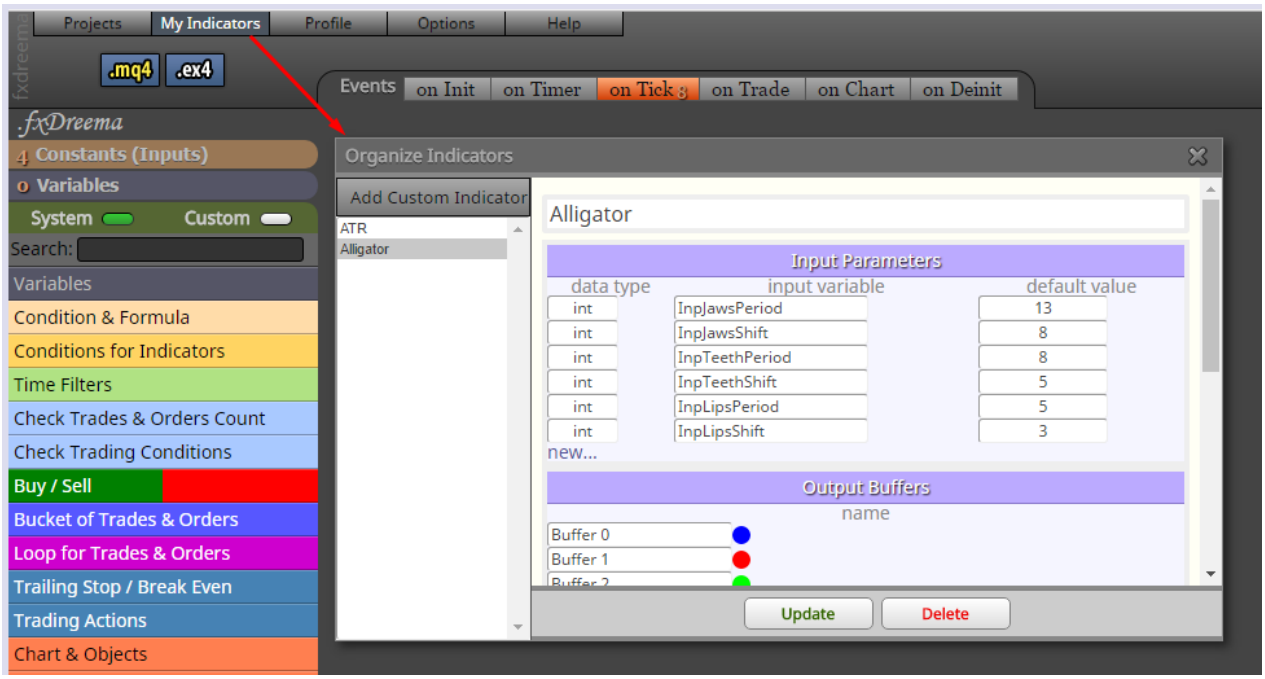
Uso de los indicadores personalizados

En fxDreema puede agregar indicadores personalizados a la base de datos para que luego puedan usarlos fácilmente en sus proyectos. Primero, vaya a **Mis indicadores.** Para agregar un nuevo indicador, necesita su archivo de código

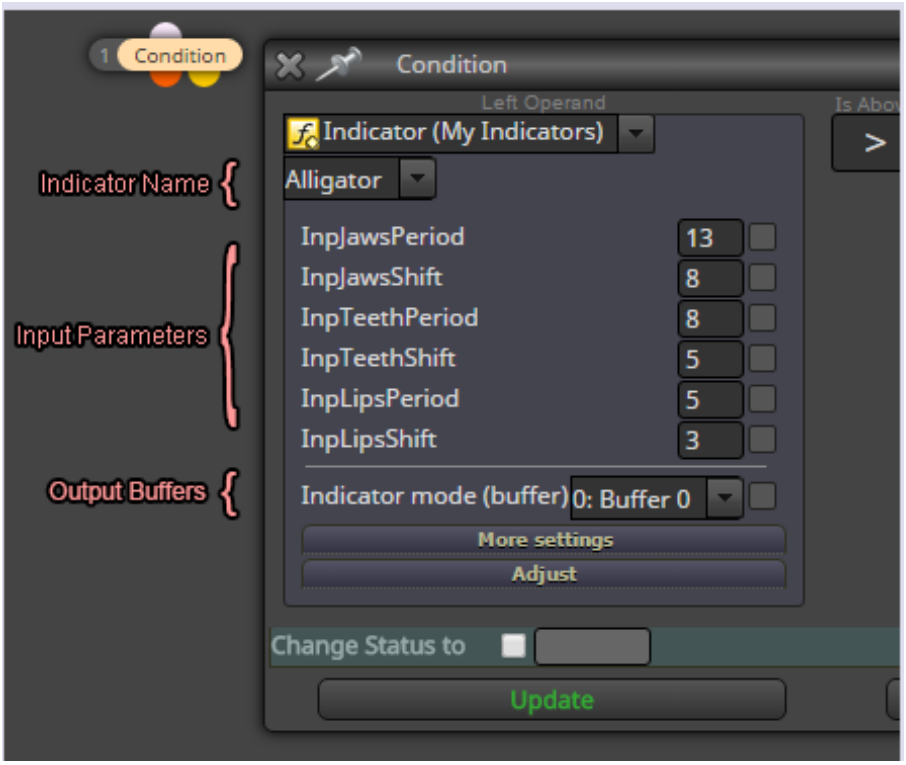
fuente, que tiene la extensión .**mq4 o .mq5.** También puede completar manualmente toda la información necesaria si no tiene el código fuente.

De hecho, el proceso de agregar un nuevo indicador desde un archivo de código fuente no implica cargar el archivo completo al servidor. Toda la información necesaria es leída localmente por el navegador y luego solo esta es enviada al servidor. Para un indicador personalizado solo necesita la siguiente información:

- **Nombre del indicador:** DEBE ser el mismo que el nombre de archivo de su indicador personalizado (sin la extensión). Si el nombre del archivo es "Custom ATR.mq5", escriba "Custom ATR".
- **Parámetros de entrada:** su recuento y orden DEBEN ser correctos; de lo contrario, el indicador personalizado no se cargará correctamente. Los nombres también deben ser los mismos, pero esto no es obligatorio. Sus tipos de datos deberían ser al menos similares.
- **Búferes de salida:** puede darles cualquier nombre que comprenda mejor.



Ahora puede usar su indicador personalizado en los bloques de fxDreema, por ejemplo, en **Condición.**



Agregar manualmente un indicador personalizado

Tiene 2 opciones: obtener la información del código fuente o de MetaTrader.

Opción 1: obtener la Información del código fuente

Puede abrir el archivo de código fuente y obtener la información desde allí. Toda la información necesaria se encuentra en algún lugar de la parte superior. Esto es parte de un indicador personalizado:

```
// + ------------------------------------------------ ------------------- +
// | Alligator.mq5 |
// | Copyright 2009-2017, MetaQuotes Software Corp. |
// | http://www.mql5.com |
// + ------------------------------------------------ ------------------- +
#property copyright "2009-2017, MetaQuotes Software Corp."
# vínculo de propiedad "http://www.mql5.com"

// ---- configuración del indicador
#property indicator_chart_window
#property indicator_buffers 3
#property indicator_plots 3
#property indicator_type1 DRAW_LINE
#property indicator_type2 DRAW_LINE
#property indicator_type3 DRAW_LINE
#property indicator_color1 Azul
#property indicator_color2 Rojo
#property indicator_color3 Lima
#property indicador_width1 1
#property indicator_width2 1
#property indicador_width3 1
#property indicator_label1 "Tiburón"
#property indicator_label2 "Dientes"
#property indicator_label3 "Labios"
//---- parámetros de entrada
input int InpJawsPeriod = 13; // Periodo de mandíbulas
input int InpJawsShift = 8; // Las mandíbulas se mueven
input int InpTeethPeriod = 8; // Periodo de los dientes
input int InpTeethShift = 5; // Cambio de dientes
input int InpLipsPeriod = 5; // Periodo de labios
input int InpLipsShift = 3; // Los labios se mueven
entrada ENUM_MA_METHOD InpMAMethod = MODE_SMMA; // Método de media móvil
input ENUM_APPLIED_PRICE InpAppliedPrice = PRICE_MEDIAN; // Precio aplicado
// ---- búfer de indicador
doble ExtJaws [];
double ExtTeeth [];
ExtLips dobles [];
// ---- asas para medias móviles
int ExtJawsHandle;
int ExtTeethHandle;
int ExtLipsHandle;
// --- barras mínimas para el cálculo
int ExtBarsMinimum;
```

Puede ver dónde se definen los parámetros de entrada:

```
input int InpJawsPeriod = 13; // Periodo de mandíbulas
input int InpJawsShift = 8; // Las mandíbulas se mueven
input int InpTeethPeriod = 8; // Periodo de los dientes
input int InpTeethShift = 5; // Cambio de dientes
input int InpLipsPeriod = 5; // Periodo de labios
input int InpLipsShift = 3; // Los labios se mueven
entrada ENUM_MA_METHOD InpMAMethod = MODE_SMMA; // Método de media móvil
input ENUM_APPLIED_PRICE InpAppliedPrice = PRICE_MEDIAN; // Precio aplicado
```

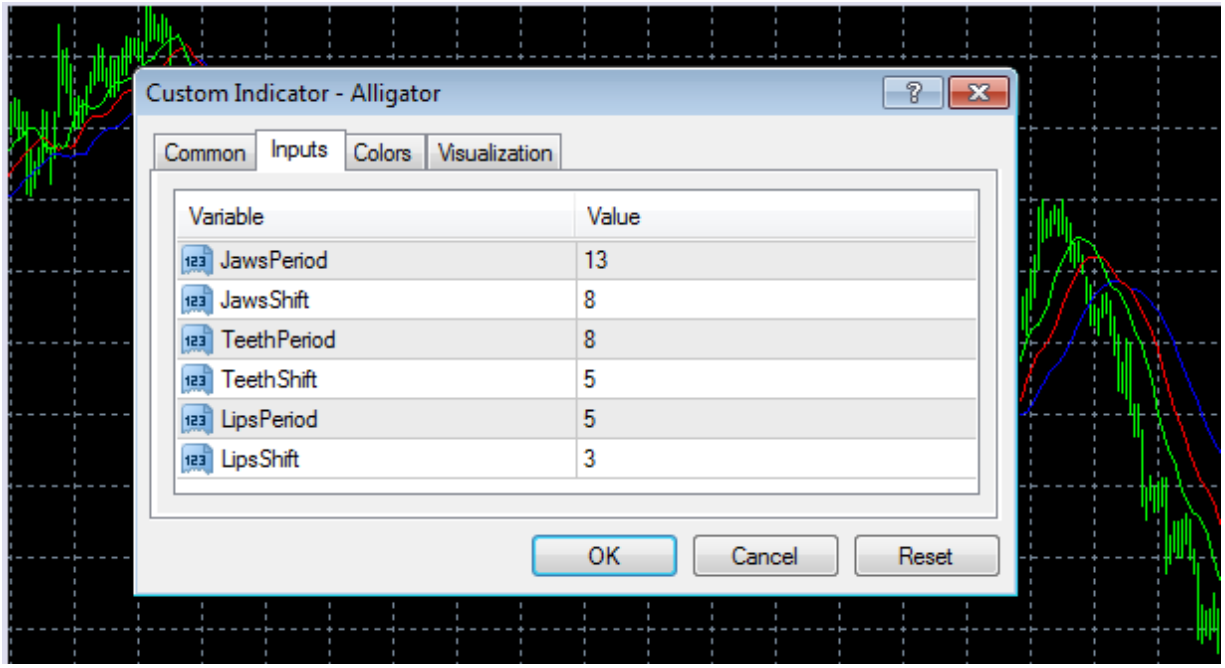
La **entrada de** palabras clave es lo que convierte estas variables globales en parámetros de entrada. La segunda palabra es el tipo de datos (int, ENUM_MA_METHOD, ENUM_APPLIED_PRICE). Luego vemos el nombre del parámetro (InpJawsPeriod, InpJawsShift ...). Después de eso tenemos los valores predeterminados (13, 8 ...). Al final tenemos la descripción de cada parámetro.

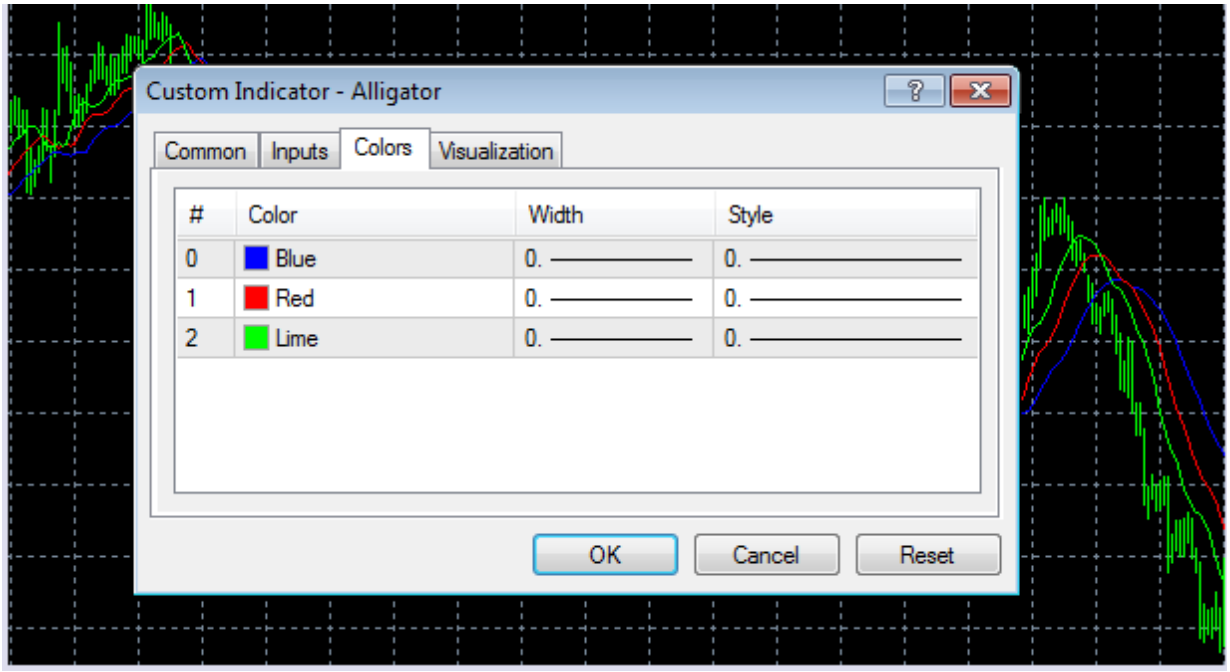
Aquí es donde podemos ver el número de búferes de salida y obviamente tenemos 3 búferes:

```
#property indicator_buffers 3
#property indicator_plots 3
```

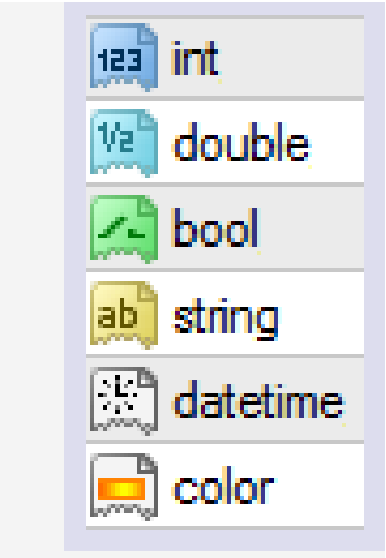
Opción 2: obtenga la Información de MetaTrader

Abra los parámetros de su indicador. Los **parámetros de entrada** se enumeran en la pestaña **Entradas** y los **búferes de salida se** encuentran en la pestaña **Colores:**





El tipo de datos de cada parámetro de entrada se presenta como un icono. Esto es lo que significa el icono:



- **int** : tipo numérico, números enteros, por ejemplo 3
- **doble:** tipo numérico, números de coma flotante, por ejemplo, 3.654
- **bool** : tipo booleano, **verdadero** o **falso**
- **cadena:** un texto, por ejemplo, "Hola, mundo"
- **fecha** y **hora:** tipo numérico, de hecho, es un valor entero, pero se acepta usar este tipo de datos para parámetros de tiempo
- **color:** tipo numérico, también un valor entero, pero diseñado para describir colores

Haga clic aquí para obtener más información sobre los tipos de datos

Constantes y variables

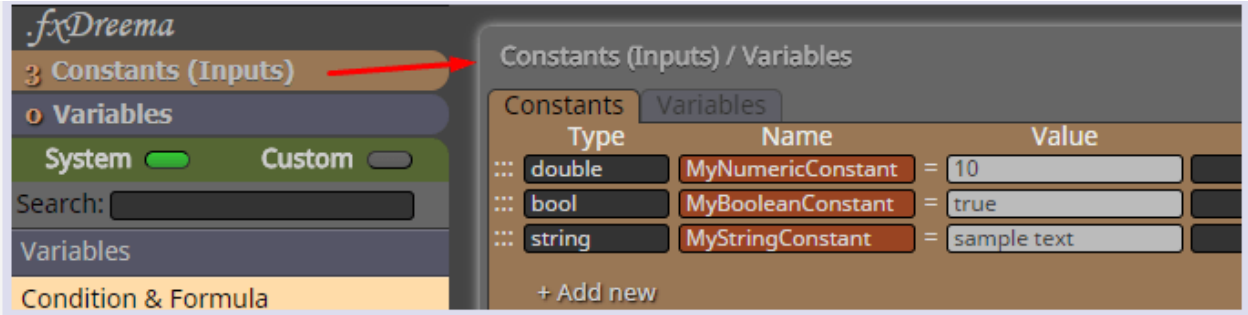
Si eres programador, sabes qué constantes y variables hay en el código. En el Builder, estas palabras significan algo un poco diferente:

- **Variables:** para el proyecto fxDreema, estas son variables globales, accesibles desde todos los bloques. En el código de salida no son exactamente variables globales.
- **Constantes:** similar a las variables, pero la idea es que no se deben modificar. Estos también son parámetros de entrada del robot. En el código de salida, en realidad no son constantes.
- **Variables terminales:** otro tipo de variables. Estas son variables globales para el terminal (MetaTrader) y en realidad se llaman **Variables globales** allí. Se utilizan para pasar información entre robots o para almacenar información durante unos días.

Constantes del proyecto (propiedades de entrada)

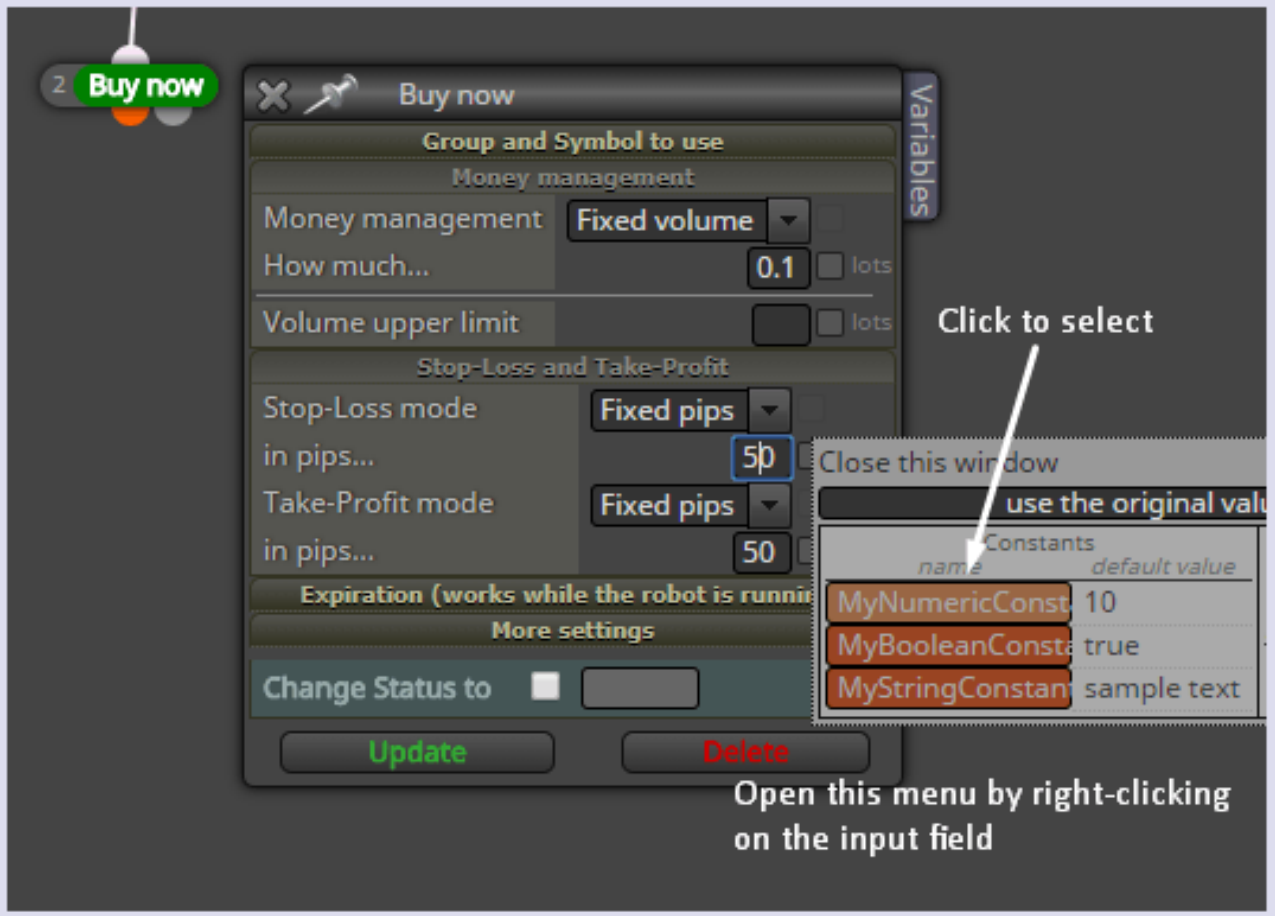
En fxDreema puede definir una sola constante y usarla en múltiples bloques. Además, esta constante también es un parámetro de entrada para el robot y también podrá utilizarla para optimizaciones.

Haga clic en **Constantes (Entradas)** para crear una nueva constante o editar Constantes ya creadas.



Una vez definidas, las constantes se pueden reorganizar.

Luego, en cualquier bloque puede seleccionar una constante para reemplazar un parámetro. Simplemente haga clic derecho en el campo de entrada que desea reemplazar:



A continuación, se muestra un ejemplo que muestra cómo se pueden usar las constantes:

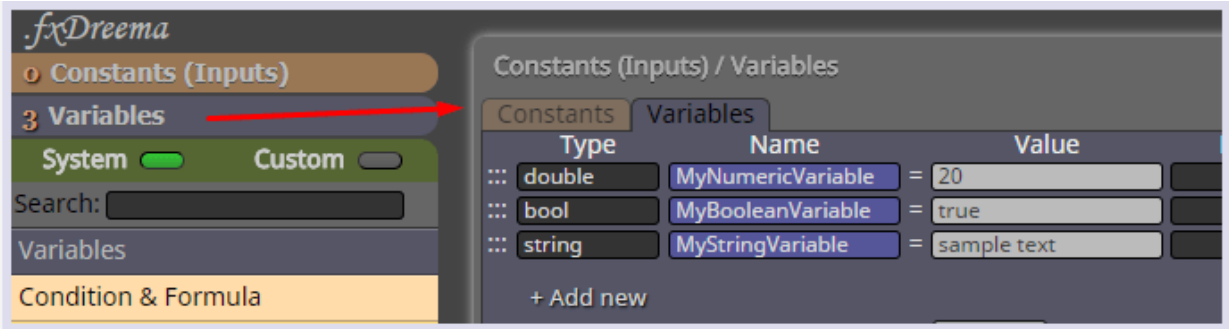
La constante única se puede utilizar en más de un bloque a la vez: las constantes son globales. Es por eso que al cambiar el valor de la constante (de los parámetros de entrada del robot), esto afecta a todos los bloques donde se usa.

No intente modificar una constante en el proyecto. La misma palabra sugiere que debería permanecer sin cambios. Si desea hacer esto, use Variables.

Variables del proyecto

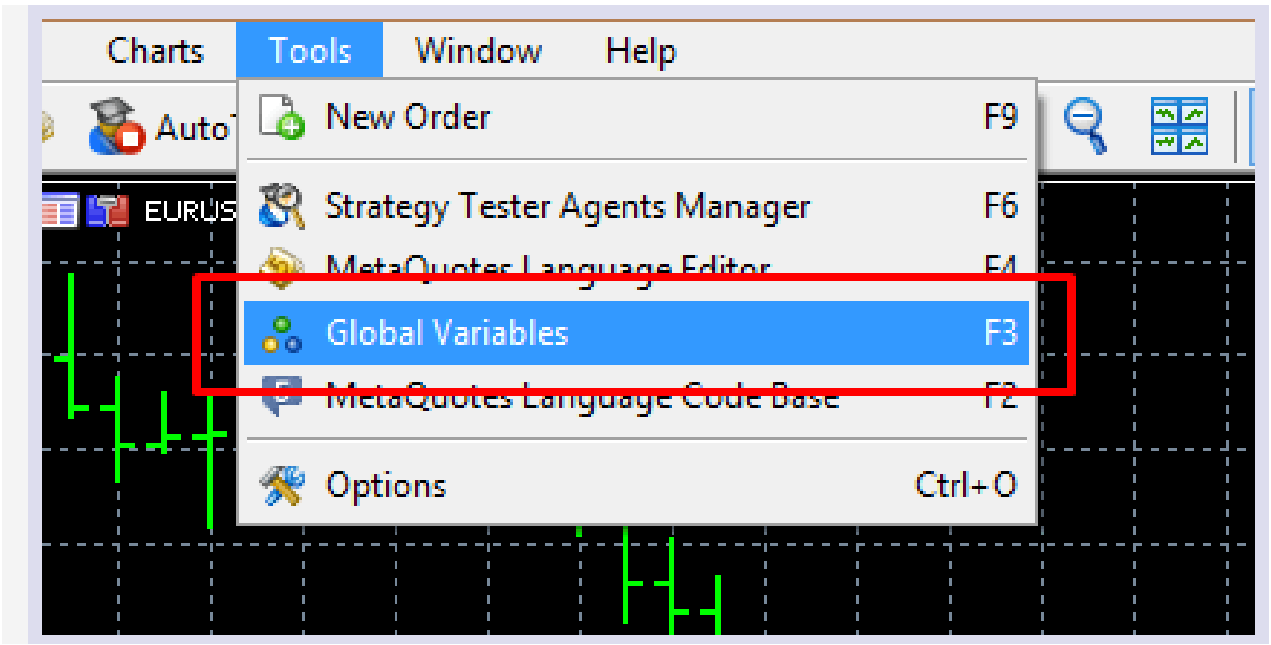
Las variables son similares, pero ligeramente diferentes a las constantes. Las Variables del Proyecto también son globales y se pueden usar de la misma forma que las Constantes, pero además sus valores se pueden modificar. Entonces, use Variables cuando desee usar un valor que pueda modificarse en cualquier bloque.

A continuación, se explica cómo definir variables. Haga clic en **Variables** para abrir esta ventana:



En los bloques puede leer y modificar Variables de varias formas. Mira el siguiente ejemplo:

Variables terminales



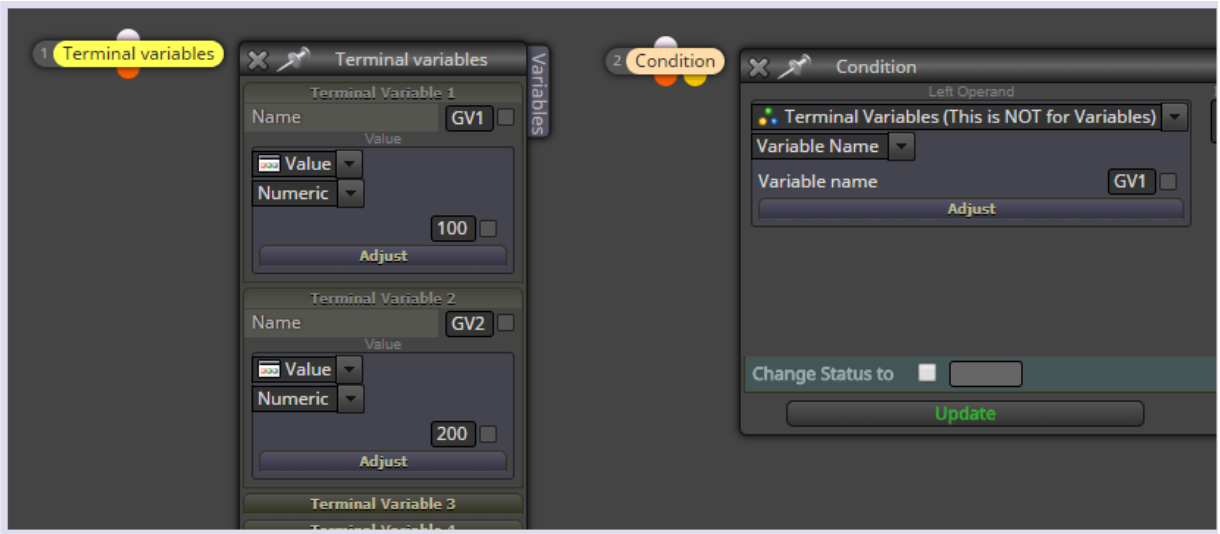
En **MetaTrader** tiene **Variables Globales,** que se llaman **Variables de Terminal** en **fxDreema** solo para reducir la confusión con los otros tipos de variables. Estas variables globales se utilizan para transferir información entre robots o como memoria persistente, debido a que se almacenan en el sistema de archivos durante 4 semanas.

Solo debe usar estas variables para almacenar valores numéricos. Las cadenas no funcionan aquí. Aquí hay más información de la documentación oficial:

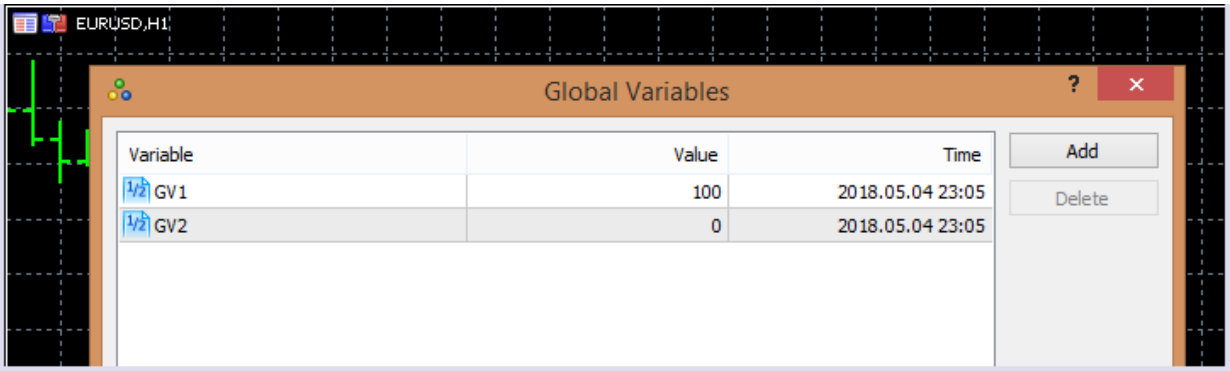
-En MetaTrader 4

-En MetaTrader 5

Puede establecer estas variables en Variables de **terminal** y obtenerlas en **Condición** (u otros bloques), como se muestra en la siguiente imagen:

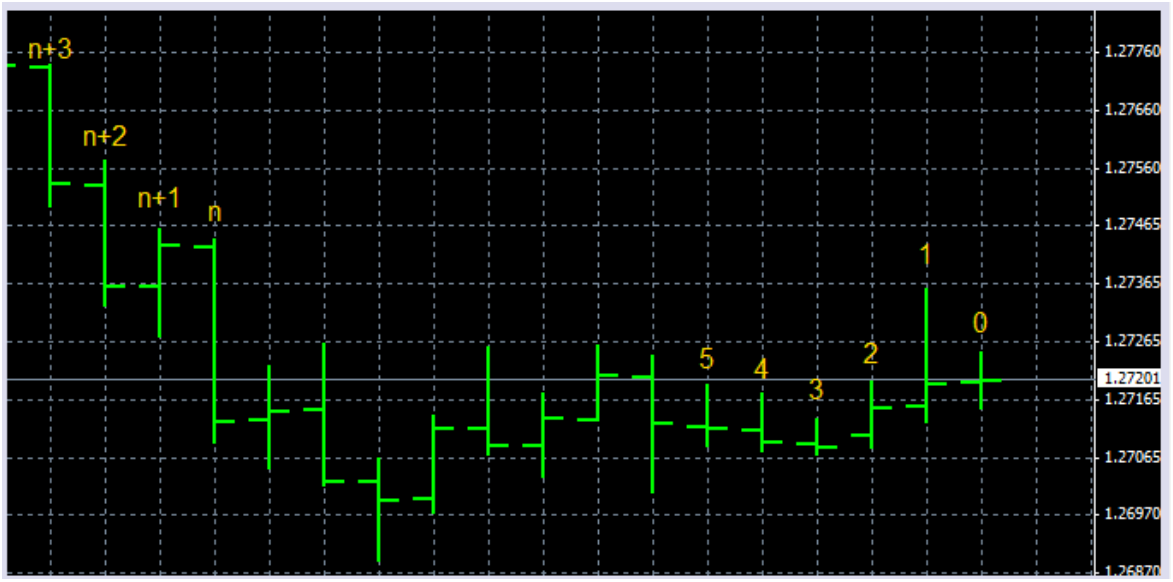


Y, por supuesto, puede ver los valores actuales y también modificarlos en **MetaTrader** cuando vaya a **Herramientas -> Variables globales:**



Resumen

La siguiente información se aplica principalmente al bloque **Condición** y sus habilidades cruzadas **× <** y **×>.** Primero veamos cómo se indexan las barras (velas) en el gráfico. Necesitamos saber esto, porque vamos a detectar cruces entre las líneas indicadoras, y esas líneas están estrechamente relacionadas con las barras.



Cada barra del gráfico tiene su propio índice (número, dirección, turno o ID, asígnele el nombre que desee). La barra más nueva del gráfico (la actual) tiene un índice **0** y el índice aumenta a medida que avanzamos en el historial. Básicamente, el número nos dice qué tan lejos de la vela actual están las velas. Lo que ahora es la vela con índice **5** será la vela **6** cuando se cree una nueva vela.

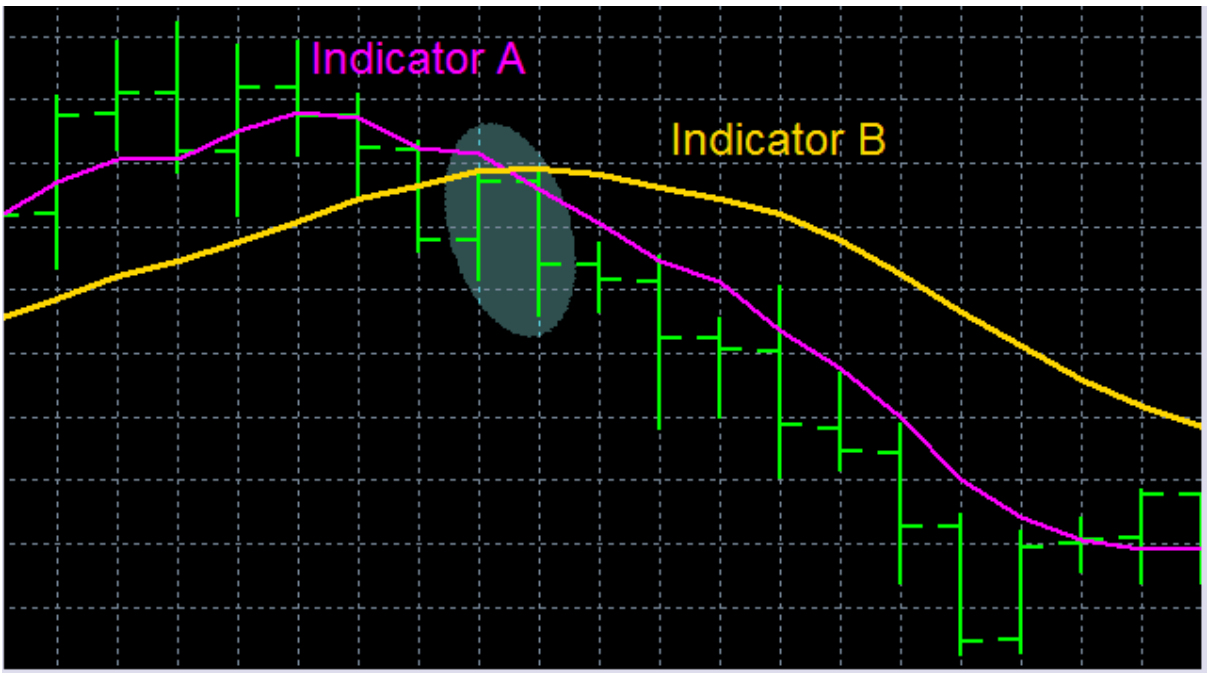
Hecho: Los valores negativos (del futuro) también son posibles y esto se puede ver en el indicador incorporado Ichimoku Kinko Hyo.

Nota: En el bloque "Condición" hay un parámetro llamado "ID de vela" (primero haga clic en "Más configuraciones") que representa el índice de la barra.

Ahora veamos cómo podemos detectar el cruce de diferentes formas.

Cruce entre dos líneas indicadoras

Si el **indicador A** y el **indicador B** son como la media móvil, así es como se cruzan:



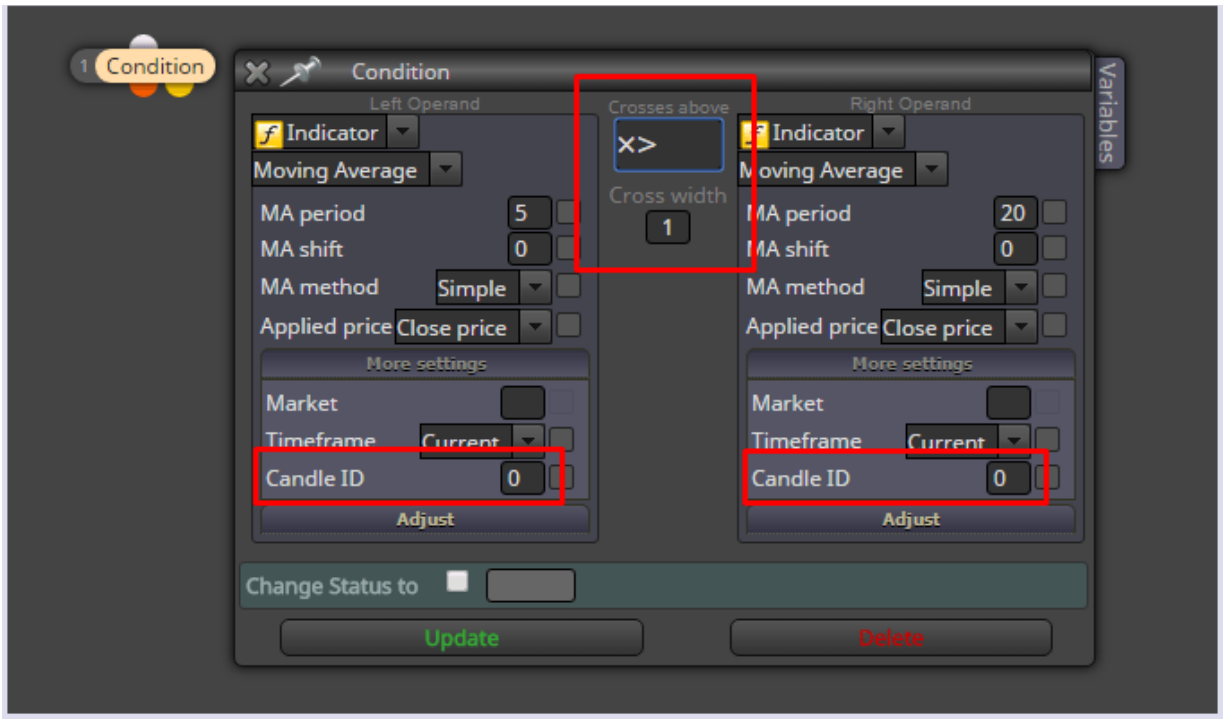
Ambas líneas indicadoras están formadas por muchos puntos, uno para cada barra del gráfico. Todos los puntos están conectados visualmente para formar esas líneas. Si amplía su gráfico lo suficiente, verá estos puntos con mucha claridad.

El cruce ocurre cuando para una barra dada, la línea A está por encima de la línea B y para su barra vecina, la línea A está debajo de la línea B. O viceversa.

-La línea A cruza la línea B por debajo cuando: **A [barra] < B [barra]** y **A [barra antigua] > B [barra antigua]**

-La línea A cruza la línea B anterior cuando: **A [barra] > B [barra]** y **A [barra antigua] < B [barra antigua]**

Así es como se ve el bloque **Condición**:



Candle ID es donde especificamos el índice de velas. Pero aunque solo podemos ver 2 indicadores, se utilizan 4 valores diferentes para detectar el cruce. Se parece a esto:

-La línea A cruza la línea B a continuación cuando: **A [ID de vela] < B [ID de vela]** y **A [ID de vela + ancho de cruz] > B [ID de vela + ancho de cruz]**

-La línea A cruza la línea B arriba cuando: **A [ID de vela] > B [ID de vela]** y **A [ID de vela + ancho de cruz] < B [ID de vela + ancho de cruz]**

Como puede ver, hay un parámetro de **ancho cruzado** adicional que le indica cuál es el índice relativo de la vela más antigua que se utiliza. Este valor es por defecto **1**.

Cruce entre la línea indicadora y un valor

¡Es posible si el valor no cambia demasiado con el tiempo!

-La línea cruza 1.066 por debajo: **Línea [barra] < 1.066** y **Línea [barra anterior] > 1.066**

-La línea cruza 1.066 arriba: **Línea [barra] > 1.066** y **Línea [barra anterior] < 1.066**

Cruce de precio vs indicador: como No hacerlo

Ahora bien, esto es complicado, porque el precio actual (Ask o Bid) es un valor que cambia con el tiempo y lo hace en pasos irregulares. Más que eso, los valores de los indicadores también cambian debido al precio. Veamos un ejemplo:

Queremos detectar el momento en que el precio Bid cruza por debajo de la línea de Media Móvil. Tenemos dos valores de indicador: el actual y el anterior (mire los puntos rojos). El precio Bid obviamente está cayendo. Ahora imagine que la oferta saltó de 1.06577 a 1.06569 directamente, sin ticks intermedios. Este movimiento es de menos de un pip y ocurre con mucha frecuencia. En el momento anterior, la oferta estaba por encima de ambos puntos indicadores y en el momento actual está por debajo de ambos. El cruce no se detectó y no se puede detectar ahora. Solo si

el precio saltó en algún lugar entre los valores de ambos indicadores, podríamos detectar un cruce, pero no hay garantía de que esto siempre suceda.

Cruce de precio vs indicador: diferentes maneras de hacerlo

Las velas capturan los movimientos de precios de alguna manera, por lo que contienen información sobre dónde estaba el precio antes y dónde está ahora. Además, los indicadores calculan sus datos utilizando información de las velas, por lo que es muy natural detectar un cruce entre una vela y una línea indicadora. Las condiciones adecuadas para comprobar aquí pueden ser las siguientes:

- Cruces de precio Línea indicadora a continuación: **Cierre de vela [bar] < Línea indicadora [bar]** y **Apertura de vela [bar] > Línea indicadora [bar]**
- Cruces de precio Línea indicadora arriba: **Vela cerrada [barra] > Línea indicadora [barra]** y **Vela abierta [barra] < Línea indicadora [barra]**

Bloques “Precio x<Indicador” y “Precio x> indicador”

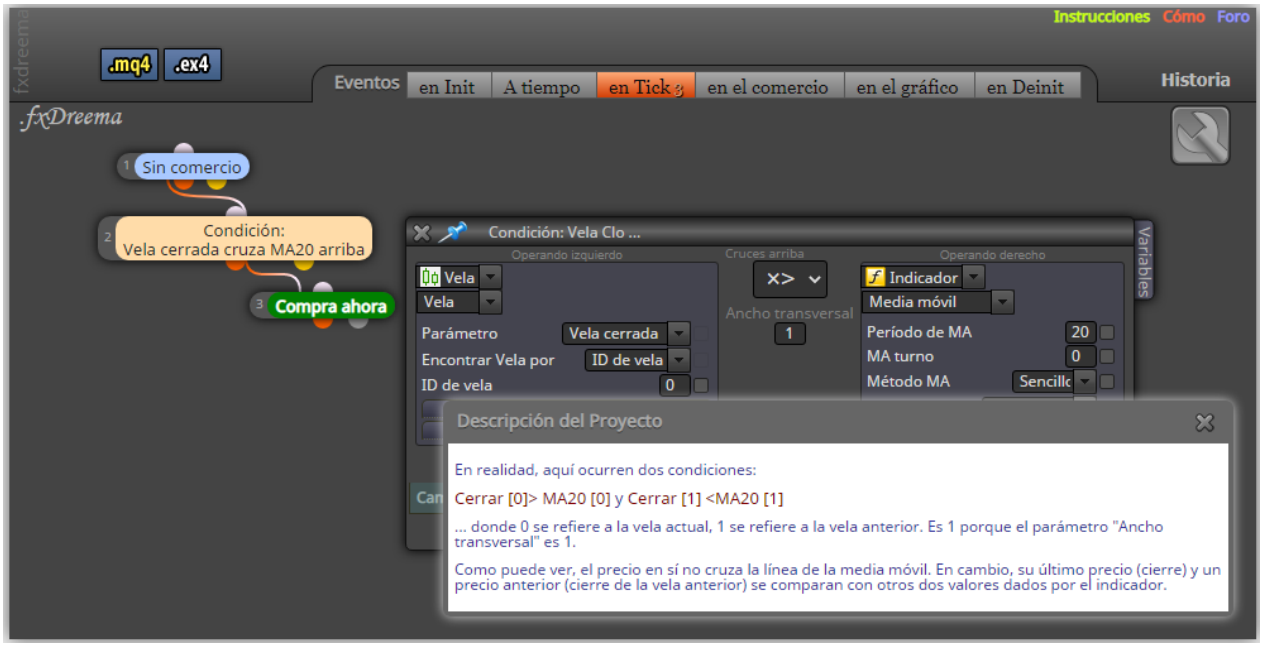
Otra forma más de detectar cruces. Estos bloques controlan el precio **Bid** y pueden detectar cuándo en el tick anterior el precio estaba en el otro lado del valor en cuestión. Muy similar al cruce anterior (con los precios de Apertura y Cierre), pero en lugar de Apertura y Cierre tenemos el precio de **Oferta** anterior y actual, por lo que estamos trabajando a un nivel "microscópico".

Tenga en cuenta que cuando esto sucede, es más como un **evento, no un estado** como en los métodos de cruce explicados anteriormente. Para los otros cruces, tenemos algo así como un estado en el que Candle Close y Candle Open están por encima y por debajo de algún valor, por ejemplo, y este estado podría ser cierto para decenas de tics por delante. Mientras estamos aquí porque estamos mirando el tick anterior y el tick actual, podemos decir que este cruce es un evento.

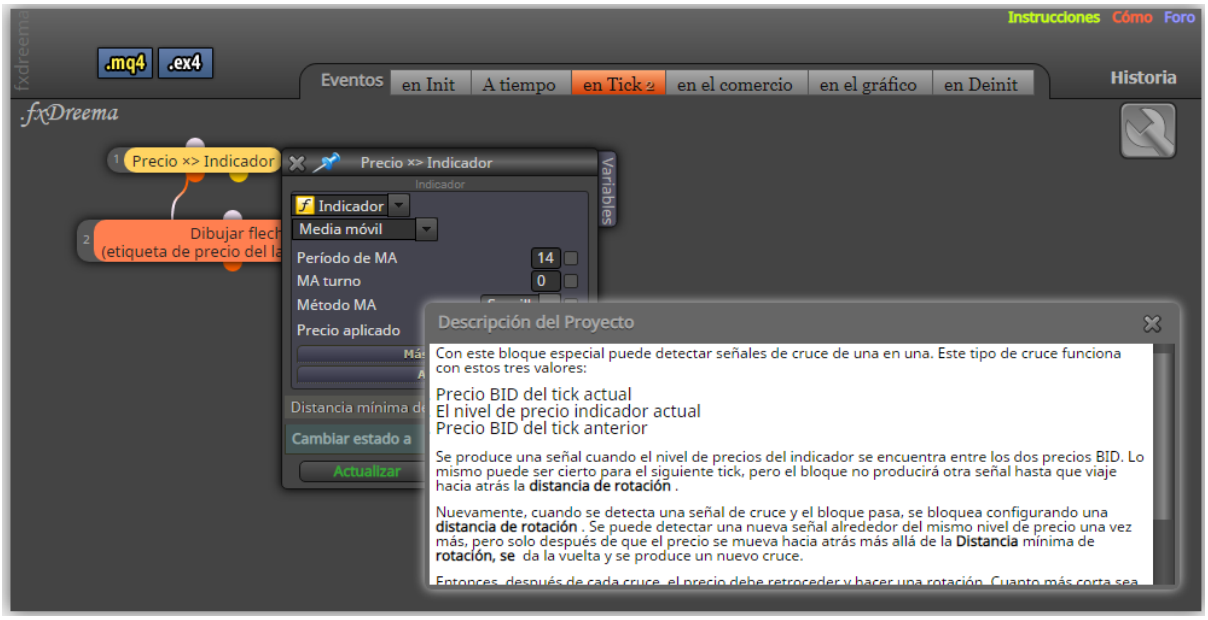
Ejemplos

Crossover: indicador vs indicador

Cruce: precio vs indicador



Cruce: precio vs indicador - método alternativo



Bucle de operaciones y órdenes

Como sabe, puede tener múltiples operaciones (o posiciones en MT5) y / o órdenes pendientes a la vez. Entonces, si desea operar con uno específico, debe señalarlo. Puede hacerlo manualmente haciendo clic en él, solo sabe dónde colocar el cursor y hacer clic con el mouse, pero su programa selecciona operaciones y órdenes de una manera ligeramente diferente.

Imagina que tienes 10 pedidos pendientes, ¿y si quieres que tu robot borre 5 de ellos, los más antiguos que fueron creados? Tenga en cuenta que el robot puede funcionar con un solo pedido a la vez. Estos son los pasos lógicos para hacerlo:

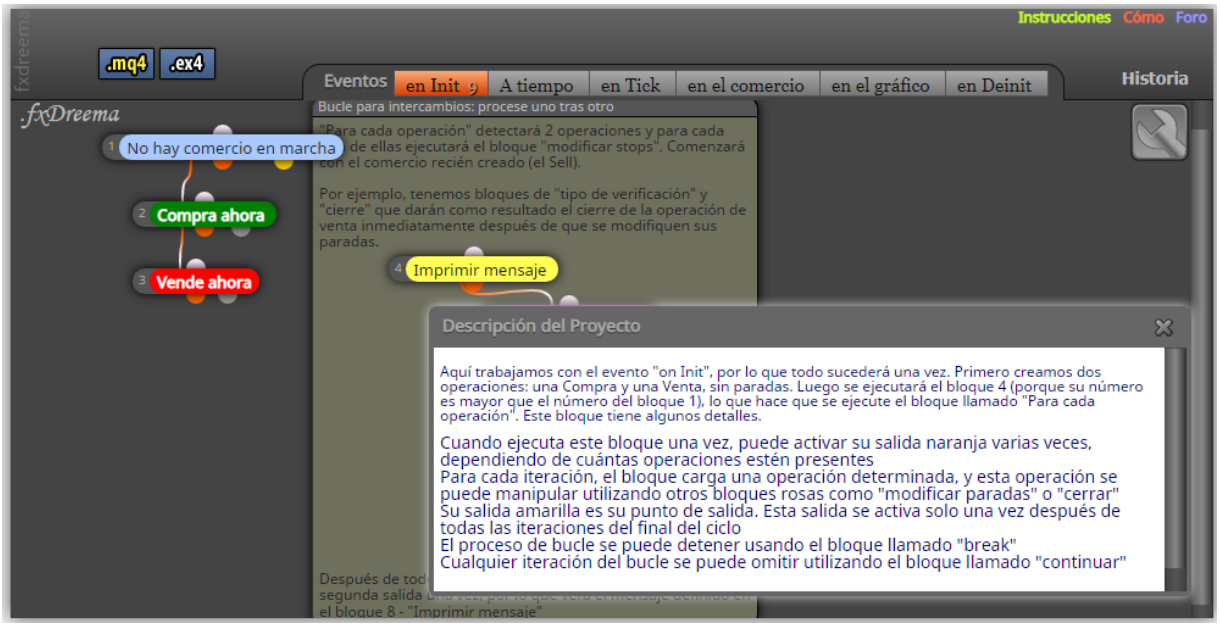
1. Verifique cuántos pedidos existen - Si no hay ninguno, no hay nada más que hacer, pero digamos que tenemos 10 pedidos, así que continuamos con el paso 2.

2. Ordenar: sabemos que tenemos 10 órdenes y conocemos sus parámetros: tiempo de apertura, tamaño del lote, tipo de negociación y muchos más. Queremos ordenar nuestros pedidos por edad.

3. Cierre 5 operaciones: clasificamos las operaciones por edad, por lo que, a partir de la más antigua, cerramos las primeras 5 que tenemos una tras otra.

Además, al pasar de un pedido a otro podemos filtrarlos, por lo que, si por ejemplo no queremos cerrar Sells, tenemos que saltarnos cualquier pedido que sea Sell y cerrar solo el que sea Buy.

El siguiente ejemplo muestra cómo trabajar con operaciones en MetaTrader 4:



Consulte otros ejemplos para ver cómo lidiar con los bucles en fxDreema: / ejemplos

Grupos y números mágicos

Número Mágico

En **MetaTrader,** el término **Número mágico** es solo una propiedad de cada operación y orden pendiente. El número mágico es **0** por defecto, pero sus Asesores Expertos pueden configurarlo en cualquier valor numérico. La idea es que cada Asesor Experto pueda reconocer sus propias operaciones y órdenes solo mirando sus **Números Mágicos**. A veces, esto es necesario, porque podría tener 2 o más Asesores Expertos realizando operaciones con el mismo nombre de **Símbolo**, y no desea que un Asesor Experto se meta con las operaciones de otro.

Por ejemplo, puede tener un Asesor Experto **A** que crea operaciones **EURUSD** con Magic Number **123** y otro Asesor Experto **B** que crea operaciones **EURUSD** con Magic Number **456**. Cuando mire la lista de operaciones en **MetaTrader** , no podrá reconocer qué operación es creada por qué **Asesor Experto**, porque los números mágicos no están listados, pero los Asesores Expertos pueden y deben hacerlo.

A veces no es necesario que te importe cuál es el número mágico. Si el Asesor Experto **A** está trabajando en **EURUSD** y el Asesor Experto **B** se coloca en **GBPUSD**, entonces el nombre del Símbolo en sí es suficiente para separar las operaciones. Ambos asesores expertos funcionarán bien y no se enredarán entre sí.

Números de grupo en fxDreema

Ya sabemos que podemos utilizar varios Asesores Expertos para trabajar con diferentes estrategias comerciales en el mismo **MetaTrader** . Pero también podemos aplicar diferentes estrategias dentro del mismo Asesor Experto, porque puede operar con operaciones con diferentes Números Mágicos. No es necesario que nuestro Asesor Experto use solo un Número Mágico, puede usar muchos.

Entonces, el número de **grupo** es solo una extensión del número mágico. Cada Asesor Experto creado con **fxDreema** tiene un parámetro de entrada llamado **MagicStart** , que es el número mágico base para todas las operaciones. El número de **grupo** es por defecto **0** (valor vacío) en todas partes, por lo que cada operación / orden tendría un **número mágico** que es igual al parámetro de entrada **MagicStart** .

Esta es la relación entre todos estos números:

Número mágico **= MagicStart + Grupo**

Ahora, imaginemos que tenemos **MagicStart** configurado como 1000. Entonces ...

- Si el grupo es 0, el número mágico será 1000
- Si el grupo es 1, el número mágico será 1001
- Si el grupo es 2, entonces el número mágico será 1002
- ... y así

Si realmente no necesita usar diferentes números de **grupo** en su proyecto, no lo cambie en los bloques. Porque si lo establece en, digamos, **1** en el bloque **Comprar ahora.** debe establecerlo en el mismo valor en todos los demás bloques involucrados con las operaciones realizadas por ese **Comprar ahora.**

Gestión del dinero en fxDreema

En los bloques diseñados para comerciar, puede especificar cómo puede cambiar el tamaño del lote con el tiempo. Por lo tanto, puede tener un pedido (en ejecución o pendiente) con diferente volumen (tamaño de lote) cada vez que se ejecutan estos bloques.

Aquí hay algunas notas:

- Si se desea un tamaño de lote inferior al mínimo permitido, se tomará el mínimo
- Si se desea un tamaño de lote mayor que el posible, se tomará el tamaño de lote más grande posible
- Hay un paso mínimo de tamaños de lote, a menudo 0.01 o 0.1 y no puede usar tamaños de lote fuera de este paso. Por ejemplo, si el paso es 0.1, solo puede abrir órdenes con lotes como 0.1, 0.2 o 0.3, pero no como 0.143. Este paso depende de su corredor.

Volumen arreglado

Ésta es la configuración predeterminada. El tamaño del lote es el que usted indique. Aunque el valor predeterminado es un número fijo, aquí también puede utilizar una fórmula variable o matemática.

% de capital => Lotes

Con esta configuración obtendrá muchos tamaños según su capital actual.

Si actualmente tiene una equidad de $ 1000, entonces con un ajuste del 100% obtendrá un tamaño de lote de 0.01 lotes cuando 1 lote equivale a $ 100,000.

Congelar % de capital (saldo, margen libre)

Con esta configuración, puede tener un tamaño de lote que reducirá el margen libre con cierta cantidad. Utilice el 40% y si tiene una equidad de $ 1000, su margen libre se reducirá en aproximadamente $ 400.

Tenga en cuenta que, debido al paso mínimo en el tamaño de los lotes, la cantidad que realmente se reduce no es exactamente la misma que la cantidad que especifique.

% De riesgo de capital (saldo, margen libre)

El tamaño del lote aquí depende del tamaño del Stop Loss. La idea es que si la operación alcanza el Stop Loss, la pérdida será aproximadamente el % del capital (saldo, margen libre) que haya especificado.

Riesgo de cantidad fija de dinero

El tamaño del lote aquí depende del tamaño del Stop Loss. Si la operación alcanza el Stop Loss, la pérdida será aproximadamente la misma que el valor que ingresó.

Ratio fijo de Ryan Jones

Con este MM, necesita obtener cierta cantidad de ganancias para aumentar el tamaño del lote. **Delta** es la cantidad de beneficio por **tamaño de unidad** para aumentar el tamaño de lote con un **tamaño de unidad**.

Como resultado, existen capas distintivas de capital en las que sus operaciones se abrirían con cierto tamaño de lote. Si comenzó con $ 1000 dólares, Tamaño de la unidad = 0.1 lote y Delta = $ 100

Unidades	Capital	Tamaño del lote
-	$ 0 - $ 1000	0,1
1	$ 1000 - $ 1100	0,1
2	$ 1100 - $ 1300	0,2
3	$ 1300 - $ 1600	0,3
4	$ 1600 - $ 2000	0.4

Obviamente, cada nivel es más grande que el anterior, así como el tamaño del lote. La siguiente capa tiene un ancho de $ 500, comienza desde $ 2000 y termina en $ 2500. Tiene un ancho de $ 500 porque 5 unidades cuestan $ 500 (5 x $ 100).

El tamaño de los lotes solo aumentaría si su capital está por encima del saldo inicial que tenía. Por debajo de eso, todos los tamaños de lote son un **tamaño de unidad**.

¡Este método de administración de dinero es global para todos los bloques! Una vez que se establezcan **Delta** y **Tamaño de la unidad**, permanecerán iguales.

Sistema de apuestas 1-3-2-6

El sistema 1-3-2-6 es un sistema de apuestas de progresión positiva. El tamaño de su lote inicial es de 1 unidad y si gana, el siguiente tamaño de lote será de 3 unidades. Si gana de nuevo - 2 unidades. Si vuelves a ganar, 6 unidades. Después

de eso, el sistema se reinicia. Después de cada pérdida, el tamaño del lote es 1 unidad y el sistema se inicia de nuevo.

También puede revertir ese sistema, lo que significa que avanzará un paso cuando pierda y reiniciará el sistema cuando gane.

Este método de administración de dinero le permite reiniciar su asesor experto sin perder información sobre su estado actual.

Sistema de apuestas D'Alembert

Es un sistema de apuestas de progresión positiva. Después de ganar, el tamaño del lote aumenta con 1 unidad y después de la pérdida se reduce con 1 unidad. Bastante simple.

También puede revertir ese sistema, por lo que el tamaño del lote aumentaría después de perder y disminuir después de ganar. Esto también se conoce como Contra D'Alembert.

Este método de administración de dinero le permite reiniciar su asesor experto sin perder información sobre su estado actual.

Sistema de apuestas Fibonacci

Este es un sistema de progresión negativa y el tamaño del lote aumenta después de la pérdida de una manera que sigue la secuencia de Fibonacci: 1, 2, 3, 5, 8, 13, 21, etc. Cada número de la secuencia es la suma de los dos anteriores. Por ejemplo, 5 sigue después de 2 y 3, porque 5 = 2 + 3.

Después de ganar

Fibonacci	Un montón	Ganar perder	Lucro	Beneficio acumulado	Nota
1	0,01	Pérdida	-10	-10	
2	0,02	Pérdida	-20	-30	
3	0,03	Pérdida	-30	-60	
5	0,05	Pérdida	-50	-110	
8	0,08	Ganar	+80	-30	En este punto se recuperan las dos últimas pérdidas
3 (retroceder 2 pasos)	0,03	Ganar	+30	0	Cubrir los gastos
1 (retroceder 2 pasos)	0,01	Ganar	+10	10	

Nota: En realidad, la secuencia de Fibonacci comienza como 0, 1, 1, 2, 3 ... pero se omiten los 0, 1 iniciales.

Este método de administración de dinero le permite reiniciar su asesor experto sin perder información sobre su estado actual.

Sistema de apuestas: Labouchére (sistema de cancelación)

Primero deberá establecer una meta de cuánto dinero le gustaría ganar. Por lo general, es más simple expresar esto en un cierto número de **unidades**, donde el valor unitario puede ser cualquier cantidad que desee. Por ejemplo, si desea ganar $ 100, puede establecer el valor unitario en $ 10, lo que significa que su objetivo es ganar 10 unidades.

Una vez que sepa cuántas unidades quiere ganar, querrá dividirlo en algunos números más pequeños y escribir esos números en una línea. Tomemos la siguiente secuencia, por ejemplo (todos los números suman 10):

1 2 3 2 2

La primera apuesta será la suma de la primera y la última unidad, o 3 unidades. Si la apuesta gana, tachamos los dos números que usamos para hacer esa apuesta. Ahora, el aspecto similar se vería así:

2 3 2

La siguiente apuesta sería de $ 40 (porque 2 + 2 = 4 unidades). Digamos que perdemos esta apuesta. Ahora, la cantidad de unidades que acabamos de perder llega al final de nuestra línea, por lo que se ve así:

2 3 4 2

Nuestra próxima apuesta ahora sería de seis unidades, o $ 60.

Para ver este sistema en su totalidad, deberá continuar haciendo apuestas hasta que haya eliminado por completo la línea. En ese momento, habrá completado el sistema Labouchére y habrá alcanzado la meta que estableció antes de comenzar.

Tenga en cuenta que si reinicia un asesor experto que trabaja con este sistema, la lista de números se restablecerá.

Apuestas: Martingala / Paroli

El sistema Martingala es bien conocido y duplica la apuesta (tamaño del lote) a la pérdida. El objetivo es ganar la cantidad de dinero de la apuesta inicial:

Apuesta	Ganar perder	Lucro	Beneficio acumulado
1	Pérdida	-10	-10
2	Pérdida	-20	-30
4	Pérdida	-40	-70
8	Pérdida	-80	-150
dieciséis	Pérdida	-160	-310
32	Ganar	+320	+10

Como puede ver, el tamaño de las apuestas aumenta exponencialmente con cada pérdida.

Lo opuesto a Martingala es Anti-Martingala, también conocido como Paroli. En este caso, duplicará el tamaño del lote cuando gane y lo reiniciará cuando pierda. También necesita reiniciar el sistema después de cierta cantidad de ganancias, de lo contrario solo perderá.

Tiene múltiples opciones, por lo que puede configurar una variedad diferente de sistemas similares de Martingala o Paroli

Este método de administración de dinero le permite reiniciar su asesor experto sin perder información sobre su estado actual.

Apuestas: secuencia personalizada

Puede definir secuencias personalizadas de unidades tanto para apostar, tanto para perder como para apostar.

Este método de administración de dinero le permite reiniciar su asesor experto sin perder información sobre su estado actual.

Stop dinámico

Importante

Trailing Stop es algo que ocurre localmente, es controlado por su Asesor Experto, no en el servidor automáticamente. Debe mantener su Asesor Experto en funcionamiento para que funcione esta funcionalidad.

En el bloque fxDreema tiene diferentes tipos de Trailing Stop, pero una cosa es en común: debe colocar su bloque Trailing Stop en algún lugar, para que pueda ejecutarse en cada tic entrante.

Funcionamiento de Traillling Stop

La idea básica es mantener el **stop-loss** de la operación a cierta distancia del precio actual cuando el precio se mueve a nuestro favor. Cuando el precio se mueve hacia atrás, el **stop loss** no cambia. Hay 2 parámetros principales: **Trailing Stop** y **Trailing Step**:

- **Trailing Stop**: este es el tamaño de **stop loss** objetivo que queremos mantener, por ejemplo, 40 pips.
- **Trailing Step**: nos indica la frecuencia **con la que** se modifica el **stop-loss** . Si **Step** es pequeño (y podría ser incluso 0), entonces tenemos modificaciones de **stop-loss** más frecuentes. Con **Step** más grande, nuestro **stop-loss se** vuelve más "perezoso", pero también hay menos solicitudes al servidor.

Ahora veamos cómo se relacionan **Trailing Stop**, **Trailing Step** y el **precio actual** (ya sea Ask o Bid). Esta es la lógica principal que ocurre para el comercio de compra:

Si (**Precio actual** - **Stop Loss**)> = (**Stop** + **Step**), mueva **Stop Loss** a (**Precio actual** - **Stop**)

Esta pseudo-fórmula no es matemáticamente precisa, porque algunos valores son absolutos y otros son relativos, pero debería obtenerla. Aquí está el significado de sus parámetros de todos modos:

- **Precio actual**: Ask o Bid. De forma predeterminada, se trata de operaciones de solicitud de compra y operaciones de oferta para venta, pero se puede cambiar.
- **Stop Loss** - El **stop-loss** nivel del comercio. No el tamaño relativo al precio actual, sino el nivel real donde se encuentra la línea de **stop-loss** .
- **Stop**: el tamaño relativo del **Trailing Stop**, por ejemplo, 40 pips.
- **Paso**: el tamaño relativo del **paso final**, por ejemplo, 1 pip.

En otras palabras, la distancia entre el precio actual y la línea de **stop-loss** debe ser al menos (Stop + Step) de ancho, por lo que el **stop-loss** se puede colocar a la distancia de Stop del precio actual.

Inicio posterior

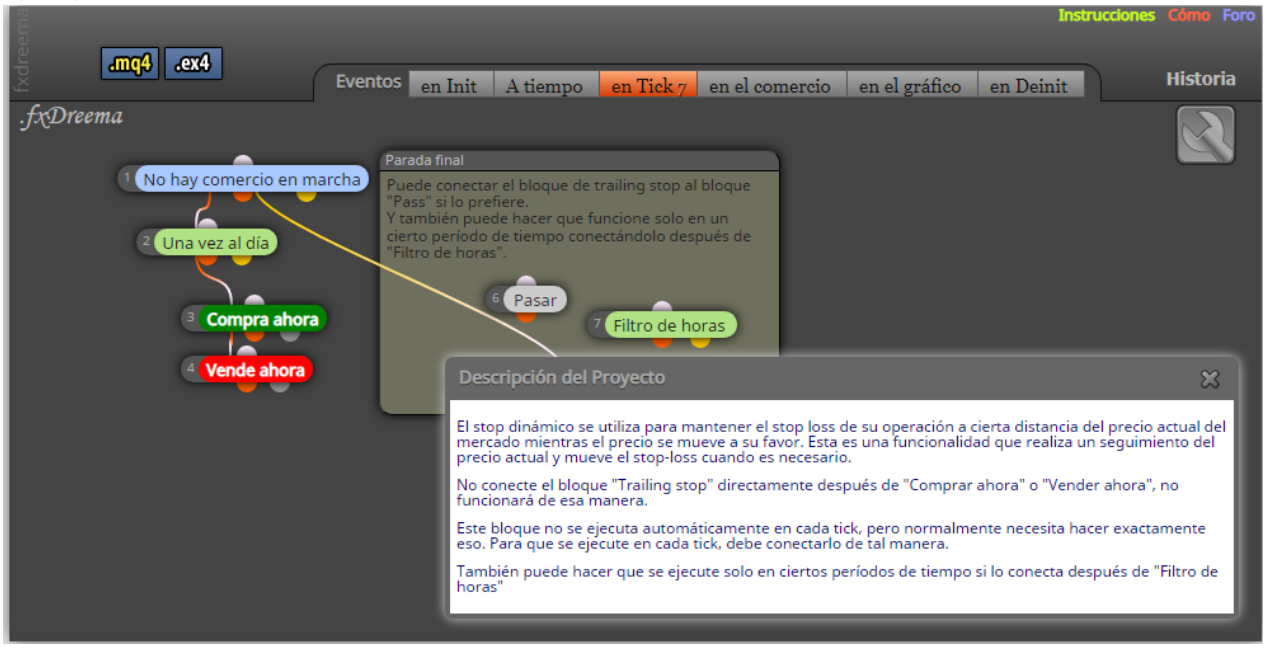
En fxDreema hay un parámetro adicional llamado **Trailing Start**. Esto crea un nivel de beneficio que debe realizarse primero, y solo entonces el bloque puede modificar la línea de **stop-loss** . Puede imaginar este nivel de umbral así:

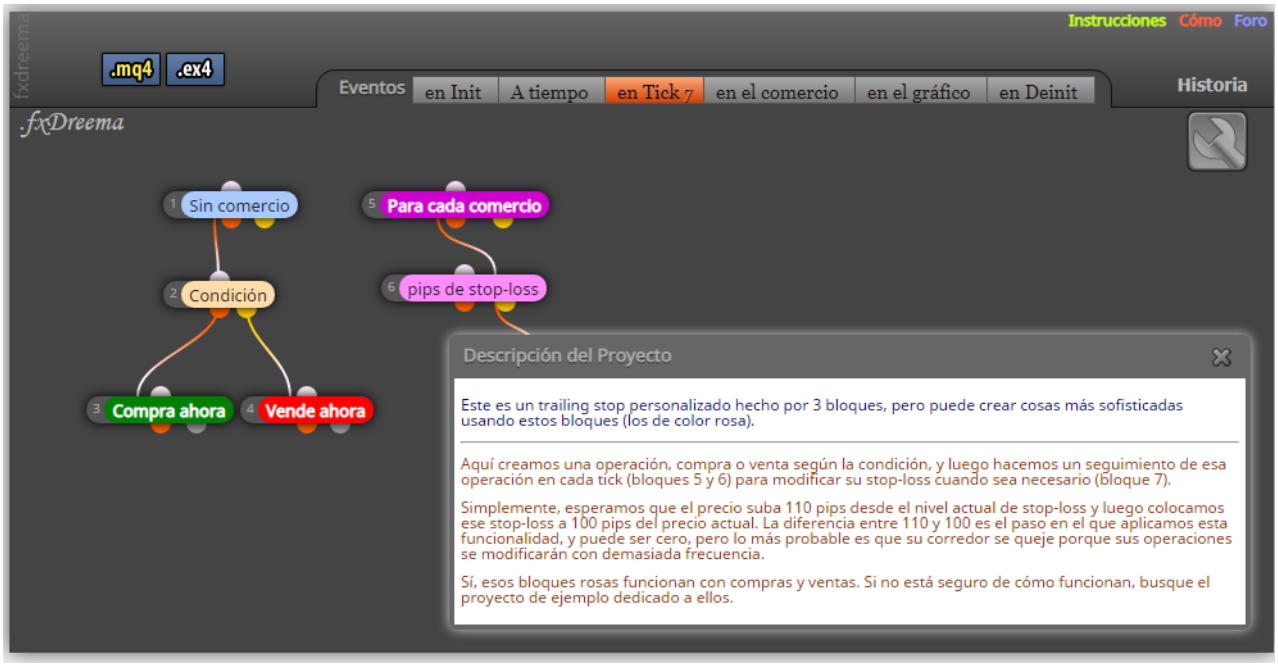
- Por encima del nivel de Trailing Start, todo funciona
- Por debajo del nivel de Trailing Start, nada funciona

Sendero

Imagine que la palabra "Stop" en Trailing Stop significa **stop-loss** o **take-profit** , porque ambos son niveles en los que se detiene la operación. En fxDreema puede rastrear no solo el **stop-loss** , sino también el **take-profit** , y hay un parámetro en el bloque Trailing Stop que puede cambiar este comportamiento.

Ejemplos





Límites de prueba y optimización

Prueba de funciones y limites en MetaTrader 4

Aquí hay un enlace donde puede ver las funciones y los límites mientras prueba y optimiza: https://www.mql5.com/en/articles/1512

Características especiales de las estrategias de prueba en datos históricos

- Algunas funciones se procesan / pasan sin salida. Estos son **Sleep ()** , **Alert ()**, **SendMail ()**, **PlaySound ()**, **MessageBox ()**, **WindowFind ()**, **WindowHandle ()** , **WindowIsVisible ()**
- El comercio está permitido solo para el símbolo bajo prueba, sin pruebas de cartera. Los intentos de operar con otro símbolo devolverán un error
- Los tamaños de lote, incluido el tamaño inicial y el paso de incremento, las comisiones y los canjes, deben tomarse de la configuración de la cuenta activa. Antes de probar, es necesario asegurarse de que haya al menos una cuenta activada en la lista de la ventana "Navegador" del terminal.
- Todos los swaps, requisitos de margen, vencimientos, órdenes GTC se modelan. Las pruebas se realizan al máximo de acuerdo con las condiciones del servidor comercial. Pero pueden producirse algunas inexactitudes en la estimación de los requisitos de margen en monedas cruzadas debido a la falta de información precisa sobre los precios de conversión en cada momento.
- La barra cero de otro período de tiempo para el mismo símbolo bajo prueba se modela aproximadamente. Abrir = correcto Abrir, Cerrar = correcto Cerrar, Bajo = mínimo (Abrir, Cerrar), Alto = máximo (Abrir, Cerrar), Volumen = Volumen final (falso)
- Se supone que el modo de ejecución instantánea se utiliza en las operaciones, y se procesa sin deslizamientos
- Procesamiento de órdenes, Abrir / Cerrar sin deslizamientos

- La prueba se detiene después de StopOut
- No se prueban los plazos semanales, mensuales e irregulares.
- La moneda del depósito se puede cambiar, pero se establecen los precios de conversión y se utilizan los disponibles actualmente
- Todavía no hay retrasos en la ejecución de las operaciones comerciales
- Está previsto introducir un retraso de configuración en el procesamiento de transacciones.
- El historial de la cuenta está completamente disponible y no depende de la configuración
- Si se utilizan activamente otros símbolos y puntos, es conveniente descargarlos con toda la profundidad posible.
- En el modelado de cada tick, el probador bombea todos los marcos de tiempo necesarios para el símbolo bajo prueba de forma independiente
- El uso de la función MarketInfo genera el error ERR_FUNCTION_NOT_ALLOWED_IN_TESTING_MODE (4059), sin embargo, se proporciona información correcta sobre los precios actuales del símbolo bajo prueba, sobre las dimensiones del nivel de parada, sobre el tamaño del punto, sobre el tamaño de propagación de cualquier símbolo que esté presente en la ventana de cotizaciones.

Características especiales del proceso de optimización

- No se muestra nada en el diario (función Print ()). Esto se hizo para acelerar las pruebas y ahorrar espacio en disco. Si se generan registros completos, los archivos de diario necesitarán cientos de MByte.
- Los objetos de dibujo no están realmente configurados. Los objetos están desactivados para acelerar la prueba.
- Se utiliza la función "Omitir resultados inútiles". Para no distorsionar la tabla y el gráfico con los resultados de las pruebas, se utiliza la posibilidad de omitir resultados muy malos. Esta función se puede habilitar en el menú contextual de "Resultados de optimización" -> pestaña "Omitir resultados inútiles".

Cosas que NO debes hacer

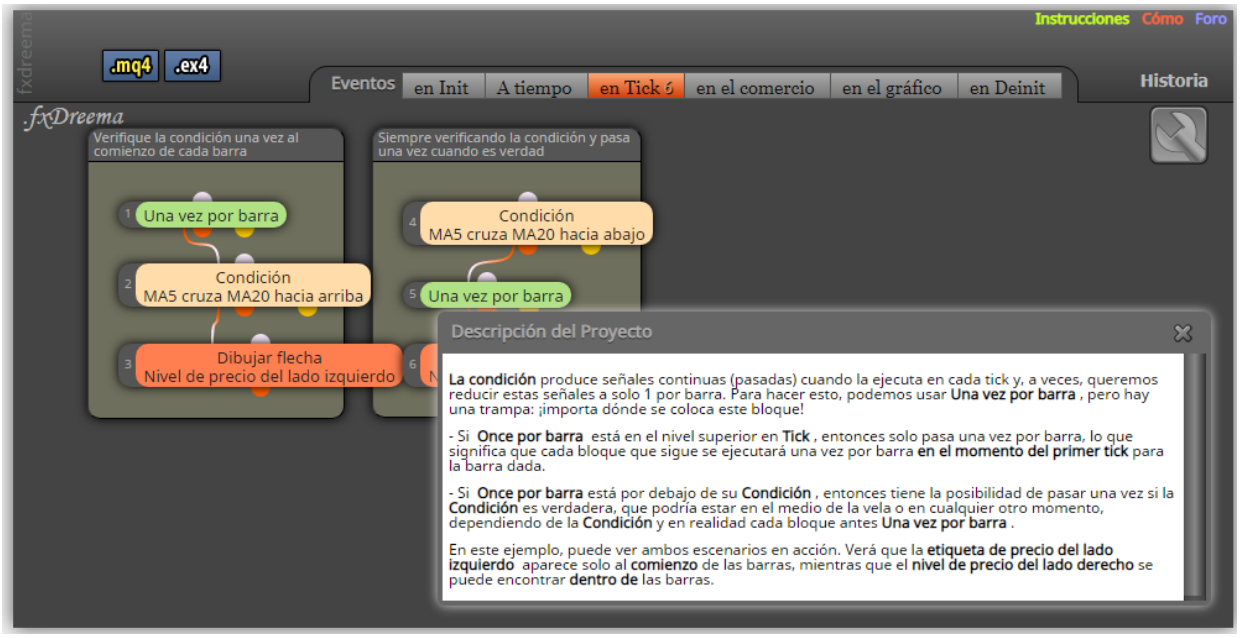
Quiero detectar cuando cierra la vela

El problema aquí es que no sabemos cuándo se cerrará la vela actual. Sí, podemos adivinar, en el último segundo, justo antes de que finalice el período de tiempo. Pero el caso es que nuestros Asesores Expertos son impulsados por tics y no sabemos cuándo llegará el último tick. Puede que no llegue 1, sino 2 segundos antes del final del período. Entonces, si no aparece ningún tic en el último segundo, ¿qué detectaremos?

Sí, tenemos el evento **Timer** y podemos configurarlo durante 1 segundo y hacer lo que queramos en el último segundo del período cada vez. Pero hay un problema: el evento **Timer** no es muy preciso en el **Probador de estrategias**. Mira, el Tester en realidad no tiene tal evento, porque no funciona en tiempo real. El Tester genera ticks falsos e impulsa al Asesor Experto de esta manera. No hay garantía de que en

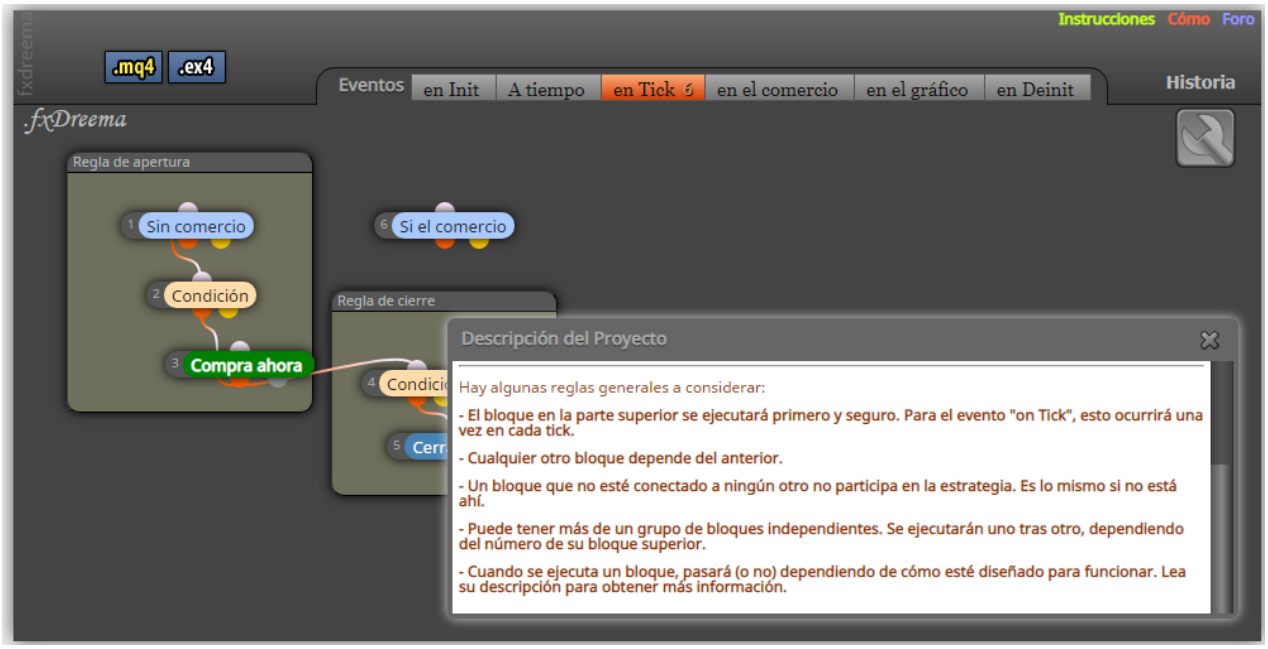
cada vela el último tic llegue en el último segundo. Por lo tanto, estamos pidiendo problemas y resultados muy inconsistentes en el Tester y en el comercio en vivo.

¿Cómo hacerlo? Es mejor usar ese bloque **Once per bar** debajo del evento **on Tick** para detectar el comienzo de cada nueva vela.



Pensando que necesitas conectar los bloques en una fila

Eche un vistazo a este proyecto y adivine qué tiene de malo:



La conexión entre los bloques 3 y 4 no debería estar allí. Puede pensar que cada bloque espera hasta que se pasa, pero este no es el caso, los bloques no se esperan unos a otros. En cambio, los bloques (al menos los del nivel superior) se ejecutan en cada tick debido al evento **on Tick** .

Bueno, en realidad hay una manera de hacer que ciertos bloques "esperen", eche un vistazo al siguiente ejemplo. Para forzar a un bloque a esperar, haga clic derecho sobre él y seleccione "Esperar para pasar". Pero esta forma de trabajar puede resultarle confusa y poco intuitiva, contrariamente a sus expectativas iniciales.

No olvide el bloque “Para cada...”
Haga clic aquí si no sabe qué significa “Para Cada”

Entonces, la categoría con los bloques rosas es un poco diferente. Los bloques "Para cada ..." están diseñados para crear un bucle de operaciones u órdenes

pendientes y el resto de los bloques rosas están diseñados para manipular una operación u orden pendiente a la vez.

La gente tiende a olvidar que en **MetaTrader 4**, y ahora también en **MetaTrader 5** podemos tener múltiples operaciones a la vez. Dado ese hecho, no puede usar, digamos, el bloque de **ganancias del cheque** rosa sin ningún contexto. Este bloque está ahí para verificar las ganancias de una operación en particular (u orden pendiente), pero no puede elegir mágicamente cuál es la correcta cuando tiene 2 o más operaciones (u órdenes) al mismo tiempo.

No coloque bloques azules en el bucle "Para cada..."

Esto no es obligatorio, pero tenga cuidado y asegúrese de saber lo que está haciendo.

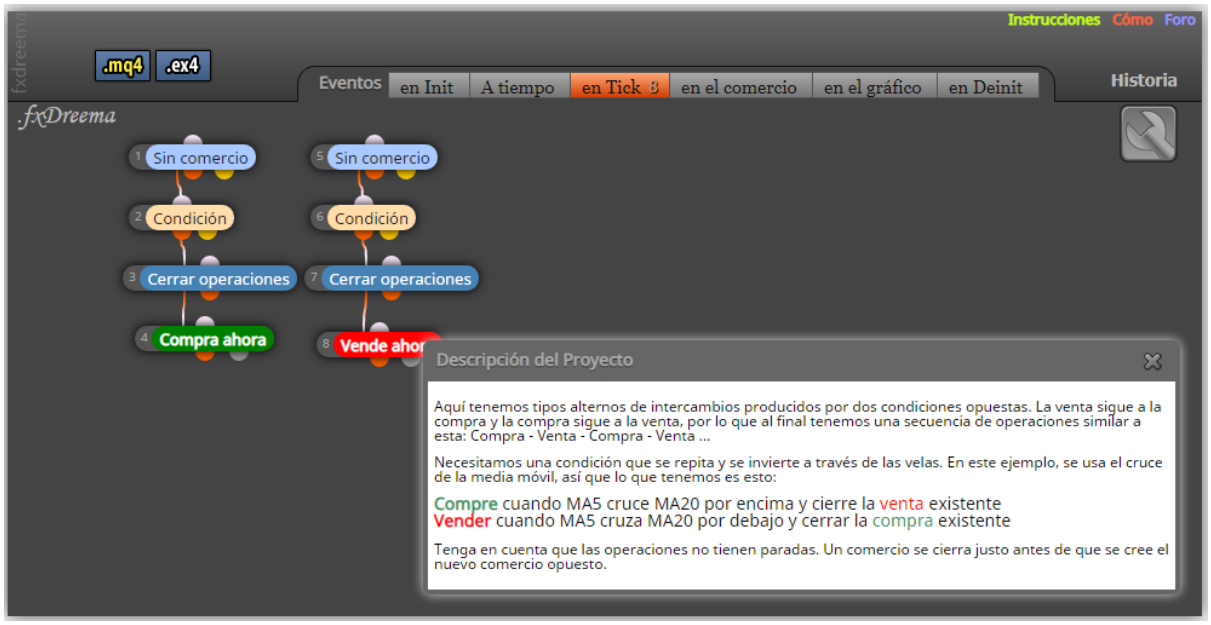
Nuevamente, un bloque "Para cada ..." crea un bucle y debajo de este bloque debe colocar otros bloques rosas, específicamente diseñados para funcionar en ese contexto. Para trabajar con una sola operación (u orden pendiente) a la vez.

No hay necesidad de conectar, digamos **Para cada Operación -> Cerrar operaciones** o tal vez **Para cada Operación -> Trailing stop**, porque esos bloques (Close trades y Trailing stop) son bloques que ya tienen bucles en ellos. En otras palabras, ya están diseñadas para encontrar todas las operaciones disponibles y hacer lo mismo para cada una de ellas, pueden trabajar solas. Mientras que el bloque de **cierre** rosa, por ejemplo, solo puede cerrar una operación (u orden pendiente) que se seleccionó previamente en un bucle realizado con uno de los bloques **For each …**

Cuidado con el bloque "Cerrar operaciones"

Cuando tenga bloques como **Comprar ahora** y **Cerrar operaciones** bajo el evento **Tick,** tenga mucho cuidado de no permitir que las **operaciones Cerrar** se ejecuten en las mismas condiciones que **Comprar ahora.** De **lo contrario, lo que sucede** es que después de que creamos una operación, se cierra inmediatamente y usted se pregunta por qué ve tantas operaciones abriéndose y cerrando.

El siguiente ejemplo es correcto y muestra cómo se pueden utilizar las **operaciones de cierre** de forma segura. Es posible que desee hacer algo diferente, pero, en cualquier caso, tenga cuidado cuando utilice este bloque y asegúrese de que se ejecute solo cuando sea necesario. Siempre debe colocar bloques como **Sin comercio** (con ciertas configuraciones) encima u otros bloques para verificar y filtrar, para limitar la posibilidad de que el bloque se ejecute en momentos inapropiados.


Instrucciones Cómo Foro
.mq4 .ex4
Eventos en Init A tiempo en Tick 8 en el comercio en el gráfico en Deinit
Historia
fxDreema
1 Sin comercio
2 Condición
3 Cerrar operaciones
4 Compra ahora
5 Sin comercio
6 Condición
7 Cerrar operaciones
8 Vende ahor
Descripción del Proyecto
Aquí tenemos tipos alternos de intercambios producidos por dos condiciones opuestas. La venta sigue a la compra y la compra sigue a la venta, por lo que al final tenemos una secuencia de operaciones similar a esta: Compra - Venta - Compra - Venta ...
Necesitamos una condición que se repita y se invierte a través de las velas. En este ejemplo, se usa el cruce de la media móvil, así que lo que tenemos es esto:
Compre cuando MA5 cruce MA20 por encima y cierre la venta existente
Vender cuando MA5 cruza MA20 por debajo y cerrar la compra existente
Tenga en cuenta que las operaciones no tienen paradas. Un comercio se cierra justo antes de que se cree el nuevo comercio opuesto.