

Consejos y Muestras de Referencia

Hay un depósito creciente de [puntas](#) de NinjaScript y muestras de referencia en <http://www.ninjatrader-support.com/vb/forumdisplay.php?f=29> el Foro de Apoyo de NinjaTrader.

-o-

Seminario: Utilización de variables de precios (el Nivel 1)

Descripción

En este seminario de nivel 1 vamos a construir un indicador de encargo que busca un modelo de barra donde la cotización de cierre es igual al precio inicial. Este indicador le mostrará como tener acceso a variables de precios y usar a un operador condicional.

-o-

Establecer

El primer paso en la creación de un indicador de encargo debe usar al mago de indicador de encargo. El mago generará el código de NinjaScript requerido que servirá de la fundación para su indicador de encargo.

1. Dentro de NinjaTrader Control Center ventana seleccionan los Instrumentos-> Nuevo NinjaScript-> Indicador... el menú 2. Presione el botón "Next>"

La definición de Propiedades de Indicador y Nombre Debajo de usted definirá su nombre de indicadores y varias propiedades de indicador.

3. Entre en la información como mostrado encima

4. Presione el botón "Next>"

p20

2

www.ninjatrader.co

m

© NinjaTrader, LLC,

2005

Definición de Parámetros de Entrada

Debajo de usted definirá sus parámetros de entrada de indicadores. Éstos son cualquier parámetro que pueda ser cambiado por el usuario y usado en el cálculo del valor (es) de indicadores. Si su indicador no requiere que cualquier parámetro deje los campos "de Nombre" en blanco.

5. Este indicador no requiere ningún parámetro de entrada 6.

Presione el botón "Next>"

La definición de Complots Debajo de usted definirá como su indicador es trazado en una carta.

p203

www.ninjatrader.com

m

© NinjaTrader, LLC,
2005

7. Entre en la información como mostrado encima. Note que cambiamos el "Color" "al Negro" y el "Tipo" a "la Barra". 8. Presione el botón "Next>"

La definición de Líneas de Oscilador Debajo de usted definirá cualquier línea de oscilador requerida. Esto sería la línea "Cero" en el indicador CCI por ejemplo. Si su indicador no requiere que cualquier línea de oscilador deje los campos "de Nombre" en blanco.

p204

www.ninjatrader.co

m

© NinjaTrader, LLC, 2005

9. Entre la información como mostrado encima (asegúrese que los campos "de Nombre" son en blanco ya que no necesitamos una línea de oscilador para un indicador de valor medio móvil simple) 10. Apriete el "Siguiete>" el botón 11. Somos terminados ahora entrando en la información establecida de nuestro indicador. Presione el botón "Finish".

* en Cualquier momento, usted puede presionar el botón "Generate" en el mago si usted no tiene que pasar por cada uno de los pasos de mago

Usted verá ahora al Redactor de NinjaScript precargado por el código de NinjaScript

generado por el mago. Debería mirar algo como la imagen abajo.

Su motivo de preocupación primario será los artículos "2" y "3".

1. Esta sección provee el indicador del nombre de demostración de carta y descripción usada en la ventana [Indicator Dialog](#). 2. Inicializar () la sección sólo es tratada una vez cuando el indicador es al principio cargado (añadido a una carta por ejemplo) y puede ser usado para establecer cualquier requisito de configuración de indicador. Usted puede ver que este mago generó el código que añadió el color de complot y diseñe esto definimos. 3. Esto es la sección que es llamada en cada actualización de barra (marca de entrada) y es donde usted entrará en su lógica de indicador

-o
-

Entrada en Lógica de Cálculo

OnBarUpdate () el método es llamado para cada marca de entrada o en el final de una barra (usuario definido) realizando cálculos de tiempo real y es llamado en cada barra de una serie de datos

p205

www.ninjatrader.com
© NinjaTrader, LLC, 2005

calculando de nuevo el indicador. Por ejemplo, un indicador sería calculado de nuevo añadiéndolo a una carta existente que hace mostrar datos de precios existentes. Por lo tanto, esto es el método principal pidió el cálculo de indicador y usaremos este método de entrar en la escritura que comprobará si el precio abierto es igual al precio cercano.

La Comparación Sustituye el código generado del mago por el código siguiente en OnBarUpdate () método en el Redactor de NinjaScript:

```
//Ponga el valor Plot0. Juego (¿Abierto [0] == Cerca [0]? 1: 0);
```

¿Aquí hacemos una comparación del precio abierto corriente `Abierto [0]` y el precio cercano corriente `Cerca [0]` utilización de un operador condicional "?" y el ajuste de un valor a la propiedad "Plot0" vía el `Juego ()` método. Podría parecer un poco intimidar pero es realmente completamente simple.

La sintaxis para un operador condicional es lo siguiente:

¿Expresión? valor 1: value2

Traducido a inglés esto dice:

¿si la expresión es verdad? use el valor 1: por otra parte use el valor 2

traducido la una vez pasada a nuestra lógica de indicador:

¿si el precio abierto corriente es igual al precio cercano corriente? use un valor de 1: otro uso sabio un valor de 0

Nuestro indicador entonces traza un valor de 1 cuando tenemos una barra con el precio abierto igual al precio cercano o un valor de 0 por otra parte. OnBarUpdate () método en su redactor debería parecer idéntico a la imagen abajo.

o-

Compilación

El código de indicador es completo ahora y tiene que ser compilado. Usted puede compilar este indicador desde dentro el Redactor de NinjaScript menú de menú "Compile" de botón de

ratón de derecho o simplemente presionar la llave F5. Es importante entender que este proceso hace el indicador listo para el uso de tiempo real y correrá natamente dentro de NinjaTrader directamente. Esto no corre interpretado tan muchas otras aplicaciones hacen. Esto le provee del rendimiento más alto posible. Si hay cualquier error hizo un informe durante la compilación, los mensajes de error serán mostrados en el fondo del Redactor de NinjaScript.

-o-

p206

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Utilización

Su estrategia está lista ahora para ser usada. Usted puede [dirigirlo en el de tiempo real](#) en seguida o [backtest esto](#).

¡Felicitaciones, usted sólo completó la construcción de una estrategia de encargo!

-o-

Seminario: Indicador en Indicador (el Nivel 2)

Descripción

En este seminario de nivel 2 vamos a construir un indicador de encargo que calcula un valor medio móvil de volumen. Este indicador le mostrará como usar el incorporado los indicadores del Valor medio móvil ([SMA](#)) y [Volumen](#).

-o-

Establecer

El primer paso en la creación de un indicador de encargo debe usar al mago de indicador de encargo. El mago generará el código de NinjaScript requerido que servirá de la fundación para su indicador de encargo.

1. Dentro de NinjaTrader Control Center ventana seleccionan los Instrumentos-> Nuevo NinjaScript-> Indicador... el menú 2. Presione el botón "Next>"

La definición de Propiedades de Indicador y Nombre Debajo de usted definirá su nombre de indicadores y varias propiedades de indicador.

3. Entre en la información como mostrado encima
4. Presione el botón "Next>"

7

p20

www.ninjatrader.co

m

© NinjaTrader, LLC,

2005

Definición de Parámetros de Entrada

Debajo de usted definirá sus parámetros de entrada de indicadores. Éstos son cualquier parámetro que pueda ser cambiado por el usuario y usado en el cálculo del valor (es) de indicadores. Si su indicador no requiere que cualquier parámetro deje los campos "de Nombre" en blanco.

5. Entre en la información como mostrado encima 6. Presione el botón "Next>"

La definición de Complots Debajo de usted definirá como su indicador es trazado en una carta.

7. Entre en la información como mostrado encima. 8. Presione el botón "Next>"

La definición de Líneas de Oscilador Debajo de usted definirá cualquier línea de oscilador requerida. Esto sería la línea "Cero" en el indicador CCI por ejemplo. Si su indicador no requiere que cualquier línea de oscilador deje los campos "de Nombre" en blanco.

9. Entre la información como mostrado encima (asegúrese que los campos "de Nombre" son en blanco ya que no necesitamos una línea de oscilador para un indicador de valor medio móvil simple) 10. Apriete el "Siguiete>" el botón 11. Somos terminados ahora entrando en la información establecida de nuestro indicador. Presione el botón "Finish".

* en Cualquier momento, usted puede presionar el botón "Generate" en el mago si usted no tiene que pasar por cada uno de los pasos de mago

Usted verá ahora al Redactor de NinjaScript precargado por el código de NinjaScript generado por el mago. Debería mirar algo como la imagen abajo.

Su motivo de preocupación primario será los artículos "2" y "3".

1. Esta sección provee el indicador del nombre de demostración de carta y descripción usada en la ventana [Indicator Dialog](#). 2. Inicializar () la sección sólo es tratada una vez cuando el indicador es al principio cargado (añadido a una carta por ejemplo) y puede ser usado para establecer cualquier requisito de configuración de indicador. Usted puede ver que este mago generó el código que añadió el color de complot y estilo y fijó el parámetro "de Período" de falta que definimos a 10. 3. Esto es la sección que es llamada en cada actualización de barra (marca de entrada) y es donde usted entrará en su lógica de indicador

-0
-

Entrada en Lógica de Cálculo

OnBarUpdate () el método es llamado para cada marca de entrada o en el final de una barra (usuario definido) realizando cálculos de tiempo real y es llamado en cada barra de una serie de datos calculando de nuevo el indicador. Por ejemplo, un indicador sería

calculado de nuevo añadiéndolo a una carta existente que hace mostrar datos de precios existentes. Por lo tanto, esto es el

p210

www.ninjatrader.com
© NinjaTrader, LLC, 2005

el método principal pidió el cálculo de indicador y usaremos este método de entrar en la escritura que calculará un valor medio móvil simple del volumen.

El cálculo del Promedio NinjaTrader ha incorporado indicadores a los cuales usted puede referirse en sus cálculos. Ya que calculamos un valor medio móvil simple del volumen tendría sentido para nosotros de usar el incorporado indicador [SMA](#) e indicadores Volume.

Sustituya el código generado del mago por el código siguiente en OnBarUpdate () método en el Redactor de NinjaScript:

```
//Calcule el promedio de volumen doble promedio = SMA (VOL (), Período) [0];
```

Aquí declaramos el "promedio" variable que es del tipo doble. Esto sirve del almacenaje temporal para el valor corriente del valor medio móvil simple del volumen. Entonces usamos el indicador de valor medio móvil simple y pase en el indicador de volumen como su entrada, pase en nuestra propiedad de indicador "Period" (un parámetro que definimos en el mago) y tenga acceso al valor corriente " [0]" que adjudicaremos a nuestro "promedio" variable. Si quisiéramos adjudicar el valor hace una barra, podríamos haber usado " [1]".

La Asignación final Entra en el código siguiente en OnBarUpdate () método y debajo del trocito de código usted entró encima:

```
//Ponga el valor deliberado al complot Plot0. Juego (promedio);
```

Aquí adjudicamos el valor "medio" a la propiedad que representa los datos de complot usando su "Juego ()" método. Acabamos de terminar de cifrar nuestro valor medio móvil simple del volumen. OnBarUpdate () método en su redactor debería parecer idéntico a la imagen abajo.

o-

-

Compilación

El código de indicador es completo ahora y tiene que ser compilado. Usted puede compilar este indicador desde dentro el Redactor de NinjaScript menú de menú "Compile" de botón de ratón de derecho o simplemente presionar la llave F5. Es importante entender que este proceso hace el indicador listo para el uso de tiempo real y correrá natalmente dentro de NinjaTrader directamente. Esto no corre interpretado como

p211

www.ninjatrader.com
© NinjaTrader, LLC, 2005

muchas otras aplicaciones hacen. Esto le provee del rendimiento más alto posible. Si hay cualquier error hizo un informe durante la compilación, los mensajes de error serán mostrados en el fondo del Redactor de NinjaScript.

-o-

Utilización

Su indicador está listo ahora para el uso y será puesto en una lista en la ventana Indicator Dialog.

Una vez aplicado a una carta, puede mirar algo como la imagen abajo.

p212

www.ninjatrader.co

m

© NinjaTrader, LLC, 2005

¡Felicitaciones, usted sólo completó la construcción de un indicador de encargo!

-o-

Seminario: Su propio SMA (el Nivel 3)

Descripción

En este seminario de nivel 3 vamos a construir un indicador de valor medio móvil simple. Este indicador le mostrará como usar "el para" el lazo y un caso solo "si" declaración.

-o-

Establecer

El primer paso en la creación de un indicador de encargo debe usar al mago de indicador de encargo. El mago generará el código de NinjaScript requerido que servirá de la fundación para su indicador de encargo.

1. Dentro de NinjaTrader Control Center ventana seleccionan los Instrumentos-> Nuevo NinjaScript-> Indicador... el menú 2. Presione el botón "Next>"

La definición de Propiedades de Indicador y Nombre Debajo de usted definirá su nombre de indicadores y varias propiedades de indicador.

p213

www.ninjatrader.co

m

© NinjaTrader, LLC,
2005

3. Entre en la información como mostrado encima 4. Hemos comprobado la opción "Complot en la carta" el que significa que esto será el indicador de revestimiento de precios (conspirará encima de los datos de precios en vez de en su propio panel de indicador) 5. Hemos comprobado la opción "Tipo de precios seleccionable" el que significa que este indicador tiene la opción de usar cualquier tipo de precios (Abierto, Alto, Bajo, Cerca etc...) 6. Presione el botón "Next>"

La definición de Parámetros de Entrada Debajo de usted definirá sus parámetros de entrada de indicadores. Éstos son cualquier parámetro que pueda ser cambiado por el usuario y usado en el cálculo del valor (es) de indicadores. Si su indicador no requiere que cualquier parámetro deje los campos "de Nombre" en blanco.

7. Entre en la información como mostrado encima 8. Presione el botón "Next>"

La definición de Complots Debajo de usted definirá como su indicador es trazado en una carta.

9. Entre en la información como mostrado encima. 10. Presione el botón "Next>"

p215

www.ninjatrader.com

© NinjaTrader, LLC,

2005

Definición de Líneas de Oscilador

Debajo de usted definirá cualquier línea de oscilador requerida. Esto sería la línea "Cero" en el indicador CCI por ejemplo. Si su indicador no requiere que cualquier línea de oscilador deje los campos "de Nombre" en blanco.

11. Entre la información como mostrado encima (asegúrese que los campos "de Nombre" son en blanco ya que no hacemos necesito una línea de oscilador para un indicador de valor medio móvil simple) 12. Apriete el "Siguiente>" el botón 13. Somos terminados ahora entrando en la información establecida de nuestro indicador. Presione el botón "Finish".

* en Cualquier momento, usted puede presionar el botón "Generate" en el mago si usted no tiene que pasar por cada uno de los pasos de mago

Usted verá ahora al Redactor de NinjaScript precargado por el código de NinjaScript generado por el mago. Debería mirar algo como la imagen abajo.

p216
www.ninjatrader.co
m
© NinjaTrader, LLC,
2005

Su motivo de preocupación primario será los artículos "2" y "3".

1. Esta sección provee el indicador del nombre de demostración de carta y descripción usada en la ventana [Indicator Dialog](#). 2. Inicializar () la sección sólo es tratada una vez cuando el indicador es al principio cargado (añadido a una carta por ejemplo) y puede ser usado para establecer cualquier requisito de configuración de indicador. Usted puede ver que este mago generó el código que añadió el color de complot y estilo, fije el parámetro "de Período" de falta que definimos a 20 y ponemos el "Revestimiento" y propiedades "PriceTypeSupported" que configuramos en el mago en los pasos 4 y 5 encima. 3. Esto es la sección que es llamada en cada actualización de barra (marca de entrada) y es donde usted entrará en su lógica de indicador

-o-

Entrada en Lógica de Cálculo

OnBarUpdate () el método es llamado para cada marca de entrada o en el final de una barra (usuario definido) realizando cálculos de tiempo real y es llamado en cada barra de una serie de datos calculando de nuevo el indicador. Por ejemplo, un indicador sería calculado de nuevo añadiéndolo a una carta existente que hace mostrar datos de precios existentes. Por lo tanto, esto es el método principal pidió el cálculo de indicador y usaremos este método de entrar en la escritura que calculará un valor medio móvil simple.

¿Hay allí bastantes barras? Sustituya el código generado del mago por el código siguiente en OnBarUpdate () método en el Redactor de NinjaScript:

p217

www.ninjatrader.co

m

© NinjaTrader, LLC, 2005

```
//No cuente si no tenemos bastantes barras si (CurrentBar <Período) vuelven;
```

Para calcular un 20 valor medio móvil de período usted necesitará mínimo de 20 barras de datos. La primera declaración en nuestro OnBarUpdate () el método comprueba para ver si hay bastantes barras de datos para realizar el cálculo de valor medio móvil. "CurrentBar" devuelve el número índice de la barra corriente y esto es comprobado contra el parámetro definido del usuario "Período". Si el número de barra corriente es menos que el usuario definió el período "volvemos" que brinco que calculan el valor medio móvil.

La adquisición de una suma de cotizaciones de cierre Entra en el código siguiente en OnBarUpdate () método y debajo del trocito de código usted entró encima:

```
//Consiga una suma de precios sobre el período de tiempo determinado dobles suma =
0; para (intervalo barsAgo = 0; barsAgo <Período; barsAgo ++) {
    suma = suma + Entrada [barsAgo];}
```

Primero debemos declarar una variable que almacenará nuestro total de suma.

dobles la suma = 0;

La "suma" variable cuyo valor es del tipo "dobles" servirá del almacenaje temporal.

```
para (intervalo barsAgo = 0; barsAgo <Período; barsAgo ++) {
    suma = suma + Entrada [barsAgo];
}
```

Después debemos calcular la suma. Usamos un estándar "para" el lazo para saltarnos por precios y añadirlos a la variable "de suma". Aunque la orden que representa el lazo pueda parecer intimidar, su realmente completamente simple. Vaya a mirarlo en inglés....

Lo que el lazo dice es:

1. el número de hace unas barras es ahora el cero
2. mientras el número de hace unas barras es menos que el período de valor medio móvil, luego vaya para rayar
3. por otra parte este lazo es terminado
4. consiga la Entrada de precios [número de hace unas barras] y añádalo al total de suma que corre
5. añada el que al número de hace unas barras (si el número de hace unas barras fuera el cero será ahora un)
6. vaya a rayar

Usted puede encontrar más información sobre [como los lazos trabajan aquí](#). Una vez que el lazo ha terminado, habrá calculado la suma total de cotizaciones de cierre para el período de nuestro valor medio móvil.

* usamos el valor de **la Entrada [barsAgo]** para conseguir que un precio use para nuestro cálculo. Podríamos haber substituido **Cerca [barsAgo]** para usar cotizaciones de cierre o **Alto [barsAgo]** para usar altos precios. La razón usamos **la Entrada [barsAgo]** consiste en ya que ponemos la propiedad "PriceTypeSupported" al verdadero. ¿Recuerde? Esto permite que el usuario del indicador seleccione un tipo de precios (Alto, Abierto, Cerca etc...) y por lo tanto pasará en el valor de tipo de precios correcto para **Introducir [barsAgo]**.

El cálculo final

p218

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Entre en el código siguiente en OnBarUpdate () método y debajo del trocito de código usted entró encima:

```
//Calcule y ponga el valor 'medio' a la propiedad 'Plot0' Plot0. El juego (suman
/ Período);
```

Nosotros podemos calcular ahora el valor de valor medio móvil final y adjudicar es el valor

a la propiedad que representa los datos de complot. Acabamos de terminar de cifrar nuestro valor medio móvil simple. OnBarUpdate () método en su redactor debería parecer idéntico a la imagen abajo.

Realización alterna En este seminario usamos "un para" el lazo para iterar a través de una colección de precios y acumular un valor de suma. Elegimos este enfoque para demostrar el uso de un lazo. Un valor medio móvil simple realmente puede ser expresado en una manera más eficiente usando el incorporado indicador [SUM](#) como el espectáculo abajo.

```
//No cuente si no tenemos bastantes barras si (CurrentBar <Período) vuelven;  
  
//Calcule y ponga el valor 'medio' a la propiedad 'Plot0' Plot0. Juego (SUMA  
(Entrada, Período) [0] / Período);
```

-o-

Compilación

El código de indicador es completo ahora y tiene que ser compilado. Usted puede compilar este indicador desde dentro el Redactor de NinjaScript menú de menú "Compile" de botón de ratón de derecho o simplemente presionar la llave F5. Es importante entender que este proceso hace el indicador listo para el uso de tiempo real y correrá natalmente dentro de NinjaTrader directamente. Esto no corre interpretado tan muchas otras aplicaciones hacen. Esto le provee del rendimiento más alto posible. Si hay cualquier error hizo un informe durante la compilación, los mensajes de error serán mostrados en el fondo del Redactor de NinjaScript.

-o-

Utilización

Su indicador está listo ahora para el uso y será puesto en una lista en la ventana Indicator Dialog.

-o
-

Seminario: Serie de Datos de Encargo Histórica (el Nivel 4)

Descripción

En este seminario de nivel 4 vamos a construir un indicador de encargo que almacena cálculos intermedios sin el uso de complots. Este indicador le mostrará como usar un objeto de DataSeries.

-o-

Establecer

El primer paso en la creación de un indicador de encargo debe usar al mago de indicador de encargo. El mago generará el código de NinjaScript requerido que servirá de la fundación para su indicador de encargo.

1. Dentro de NinjaTrader Control Center ventana seleccionan los Instrumentos-> Nuevo NinjaScript-> Indicador... el menú 2. Presione el botón "Next>"

La definición de Propiedades de Indicador y Nombre Debajo de usted definirá su nombre de indicadores y varias propiedades de indicador.

p220

www.ninjatrader.co

m

© NinjaTrader, LLC,
2005

3. Entre en la información como mostrado encima 4. Apriete el "Siguiete>" el botón 5. Presione el botón "Next>" otra vez

La definición de Parámetros de Entrada Debajo de usted definirá sus parámetros de entrada de indicadores. Éstos son cualquier parámetro que pueda ser cambiado por el usuario y usado en el cálculo del valor (es) de indicadores. Si su indicador no requiere que cualquier parámetro deje los campos "de Nombre" en blanco.

p221

www.ninjatrader.co

m

© NinjaTrader, LLC, 2005

6. Entre en la información como mostrado encima 7. Presione el botón "Next>"

La definición de Complots Debajo de usted definirá como su indicador es trazado en una carta.

8. Entre en la información como mostrado encima 9. Presione el botón "Next>"

La definición de Líneas de Oscilador Debajo de usted definirá cualquier línea de oscilador requerida. Esto sería la línea "Cero" en el indicador CCI por ejemplo. Si su indicador no requiere que cualquier línea de oscilador deje los campos "de Nombre" en blanco.

10. Entre la información como mostrado encima (asegúrese que los campos "de Nombre" son en blanco ya que no hacemos necesito una línea de oscilador para un indicador de valor medio móvil simple) 11. Apriete el "Siguiente>" el botón 12. Somos terminados ahora entrando en la información establecida de nuestro indicador. Presione el botón "Finish".

* en Cualquier momento, usted puede presionar el botón "Generate" en el mago si usted no tiene que pasar por cada uno de los pasos de mago

Usted verá ahora al Redactor de NinjaScript precargado por el código de NinjaScript generado por el mago. Debería mirar algo como la imagen abajo.

Su motivo de preocupación primario será los artículos "2" y "3".

1. Esta sección provee el indicador del nombre de demostración de carta y descripción usada en la ventana Indicator Dialog. 2. Inicializar () la sección sólo es tratada una vez cuando el indicador es al principio cargado (añadido a una carta por ejemplo) y puede ser usado para establecer cualquier requisito de configuración de indicador. Usted puede ver que este mago generó el código que añadió el color de complot y diseñe esto definimos. 3. Esto es la sección que es llamada en cada actualización de barra (marca de entrada) y es donde usted entrará en su lógica de indicador

-0-

Entrada en Lógica de Cálculo

OnBarUpdate () el método es llamado para cada marca de entrada o en el final de una barra (usuario definido) realizando cálculos de tiempo real y es llamado en cada barra de una serie de datos calculando de nuevo el indicador. Por ejemplo, un indicador sería calculado de nuevo añadiéndolo a una carta existente que hace mostrar datos de precios existentes. Por lo tanto, esto es el método principal pidió el cálculo de indicador y usaremos este método de entrar en la escritura que hará nuestros cálculos.

La creación de Objeto de DataSeries 1. Declare una variable ("myDataSeries" usada en este ejemplo) del tipo DataSeries que sostendrá un objeto de DataSeries dentro de la región "Variables" 2. Cree un nuevo objeto de DataSeries y assign esto a la variable de DataSeries

dentro de Inicializar () método

p224
www.ninjatrader.com
© NinjaTrader, LLC,
2005

```
# Variables de región
    DataSeries privado myDataSeries; //Declaran una variable de DataSeries #
endregion

protegido anulan el vacío Inicializan () {
    myDataSeries = nuevo DataSeries (esto); //esto se refiere al
    indicador/estrategia sí mismo
                                     //y sincroniza el objeto de
DataSeries a
histórico
                                     //barras de datos}
```

El almacenaje de cálculos en DataSeries se opone Ahora que hacemos nuestro DataSeries objetar a que podamos almacenar dobles valores en ello. Para este ejemplo almacenaremos un Final simple menos el cálculo Abierto.

Entre en el código siguiente en OnBarUpdate () método:

```
//Calcule la variedad de la barra corriente y ponga el valor myDataSeries. Juego
(Cerca [0] - Abierto [0]);
```

El valor de un objeto de DataSeries será alineado con la barra corriente. Esto significa que todos los objetos de DataSeries serán sincronizados con el índice de CurrentBar. Esto permite que usted almacene un doble valor que corresponde a cada barra.

La utilización de valores de DataSeries Con nuestro nuevo DataSeries objeta a que podamos seguir con cálculos adicionales fácilmente. Podemos usar ahora nuestro objeto de DataSeries tan entrada a un método de indicador, como el SMA o en vez de escribir siempre Cerca [0] - Abierto [0] podemos subsitute nuestro objeto de DataSeries en cambio según el ejemplo abajo.

Para trazar nuestro cálculo final almacenaremos el cálculo en nuestro Plot0. 1. Quite el código generado del mago en OnBarUpdate () de "Plot0. Juego (Cerca [0]);" 2. En OnBarUpdate () método añaden el trocito de código siguiente

```
//Calcule la variedad de la barra corriente y ponga el valor Plot0. Juego (SMA
(SMAPeriod) [0] + myDataSeries [0]);
```

Aquí adjudicamos el SMA + DataSeries valoran a la propiedad que representa los datos de complot usando su "Juego ()" método. Acabamos de terminar de cifrar nuestro ejemplo de DataSeries.

-o-

Compilar

El código de indicador es completo ahora y tiene que ser compilado. Usted puede compilar este indicador desde dentro el Redactor de NinjaScript menú de menú "Compile" de botón de ratón de derecho o simplemente presionar la llave F5. Es importante entender que este proceso hace el indicador listo para el uso de tiempo real y correrá natalmente dentro de NinjaTrader directamente. Esto no corre interpretado tan muchas otras aplicaciones hacen. Esto le provee del rendimiento más alto posible. Si hay cualquier error hizo un informe durante la compilación, los mensajes de error serán mostrados en el fondo del Redactor de NinjaScript.

p225

www.ninjatrader.com
© NinjaTrader, LLC, 2005

-o-

Utilización

Su indicador está listo ahora para el uso y será puesto en una lista en la ventana Indicator Dialog.

Una vez aplicado a una carta, puede mirar algo como la imagen abajo.

p226

www.ninjatrader.com
© NinjaTrader, LLC, 2005

¡Felicitaciones, usted sólo completó la construcción de un indicador de encargo!

-o-

Seminario: Colores de Complot de Encargo vía Umbrales (el Nivel 5)

Descripción

En este seminario de nivel 5 vamos a construir un indicador de encargo que es una variación ROC y pinta un color encima del grupo cero y el otro abajo. Este indicador le mostrará como usar el concepto de umbrales de complot.

-o-

Establecer

El primer paso en la creación de un indicador de encargo debe usar al mago de indicador de encargo. El mago generará el código de NinjaScript requerido que servirá de la fundación para su indicador de encargo.

1. Dentro de NinjaTrader Control Center ventana seleccionan los Instrumentos-> Nuevo NinjaScript-> Indicador... el menú 2. Presione el botón "Next>"

La definición de Propiedades de Indicador y Nombre Debajo de usted definirá su nombre de indicadores y varias propiedades de indicador.

3. Entre en la información como mostrado encima
4. Presione el botón "Next>"

5. Este indicador requiere dos parámetros "Período" y "Liso", entre en la información como mostrado encima 6. Presione el botón "Next>"

La definición de Complots Debajo de usted definirá como su indicador es trazado en una carta.

7. Entre en la información como mostrado encima. Note que hemos añadido dos complots. El complot "de AboveZero" será trazado si el valor de indicador está encima del cero y el

complot "de BelowZero" será trazado si el indicador es bajo cero. Todavía tenemos que añadir el código de NinjaScript para hacer esto posible que viene más tarde al seminario. 8. Presione el botón "Next>"

La definición de Líneas de Oscilador Debajo de usted definirá cualquier línea de oscilador requerida. Esto sería la línea "Cero" en el indicador CCI por ejemplo. Si su indicador no requiere que cualquier línea de oscilador deje los campos "de Nombre" en blanco.

9. Entre en la información como mostrado encima. Añadimos una línea de oscilador que será trazada en un valor del cero. 10. Apriete el "Siguiete>" el botón 11. Somos terminados ahora entrando en la información establecida de nuestro indicador. Presione el botón "Finish".

* en Cualquier momento, usted puede presionar el botón "Generate" en el mago si usted no tiene que pasar por cada uno de los pasos de mago

Usted verá ahora al Redactor de NinjaScript precargado por el código de NinjaScript generado por el mago. Debería mirar algo como la imagen abajo.

Su motivo de preocupación primario será los artículos "2" y "3".

1. Esta sección provee el indicador del nombre de demostración de carta y descripción usada en la ventana [Indicator Dialog](#). 2. Inicializar () la sección sólo es tratada una vez cuando el indicador es al principio cargado (añadido a una carta por ejemplo) y puede ser usado para establecer cualquier requisito de configuración de indicador. Usted puede ver que este mago generó el código que añadió el color de complot y diseñe esto definimos. 3. Esto es la sección que es llamada en cada actualización de barra (marca de entrada) y es donde usted entrará en su lógica de indicador

-o-

Entrada en Lógica de Cálculo

OnBarUpdate () el método es llamado para cada marca de entrada o en el final de una barra (usuario definido) realizando cálculos de tiempo real y es llamado en cada barra de una serie de datos calculando de nuevo el indicador. Por ejemplo, un indicador sería calculado de nuevo añadiéndolo a una carta existente que hace mostrar datos de precios existentes. Por lo tanto, esto es el método principal pidió el cálculo de indicador y usaremos este método de entrar en la escritura que calculará el valor de ROC.

Estableciendo Umbrales de Complot Inicializar () el método es llamado una vez antes de que cualesquiera datos de barra sean cargados y sean usados para configurar el indicador. El código abajo es automáticamente generado por el mago y añadido a Inicializar () método. Esto configura el indicador para dos complots y una línea y fija los parámetros.

p230

www.ninjatrader.com
© NinjaTrader, LLC, 2005

```
Complots. Añada (nuevo Complot (Color. Verde, PlotStyle. Línea, "AboveZero"));  
Complots. Añada (nuevo Complot (Color. OrangeRed, PlotStyle. Línea, "BelowZero"));  
Líneas. Añada (nueva Línea (Color. Negro, 0, "ZeroLine")); Período = 14; Liso = 3;
```

Entre en el código siguiente en Inicializar () el método y debajo del mago generó el código:

```
//Ponga los valores umbrales para cada complot Complots [0].Min = 0; Complots [1].Max  
= 0;
```

El concepto de poner valores umbrales debe ponerse cuando y cuando no pintar un complot en la carta. En este indicador, tenemos un complot "de AboveZero" con un color de falta del verde que sólo queremos ver cuando el valor de ROC está encima del cero y un complot "de BelowZero" con un color de falta del de color zanahoria que sólo queremos ver cuando el valor de ROC es bajo cero. A fin de hacer lo que pasa tenemos que poner los valores umbrales de cada complot.

```
Complots [0].Min = 0;
```

Esta declaración dice, en la colección de objetos de Complot, tome el primer (Complots [0]) y ponga su valor mínimo al cero. Esto significa que cualquier valor bajo cero no mostrará.

```
Complots [1].Max = 0;
```

Esta declaración dice, en la colección de objetos de Complot, tome el segundo (Complots [1]) y ponga su valor máximo al cero. Esto significa que cualquier valor encima del cero no mostrará.

Ahora tenemos un mecanismo de conmutación de complot simple que muestra la línea coloreada correcta según si el valor de ROC está encima o bajo cero. De hecho, usted puede tomar este concepto un poquito más lejos. Usted puede poner hasta el estilo de complots diferente (barra, punto etc.) según valores umbrales.

Una palabra rápida sobre colecciones. Las colecciones son objetos que almacenan una colección de objetos, la clase del parecido un contenedor. En este caso trabajamos con una colección de complots. En el código generado del susodicho mago usted notará que añadimos nuevos complots a la colección "de Complots". "AboveZero" fue añadido primero y

luego "BelowZero". Esto significa que podemos referirnos al objeto de complot "de AboveZero" por Complots [0]. La razón que no pasamos en un valor de 1 consiste en porque las colecciones son índices basados cero. Esto significa que el primer artículo tiene un índice de 0, la segunda vez un índice de 1 etcétera.

Completar el indicador Indicator This es realmente completamente simple en su realización. La última cosa que tenemos que hacer es añade el código de cálculo y puso el valor de ROC a ambas nuestras líneas de complot. Sustituya el código generado del mago por el código siguiente en OnBarUpdate () método en el Redactor de NinjaScript:

```
//Hay allí bastantes barras si (CurrentBar <Período) vuelven;  
  
//Póngase el complot valora AboveZero. Juego (ROC (Período) [0]); BelowZero. Juego  
(ROC (Período) [0]);
```

El cálculo primero comprueba para asegurar que hay bastantes barras para completar el cálculo y luego pone ambas líneas de complot al valor de ROC.

OnBarUpdate () método en su redactor debería parecer idéntico a la imagen abajo.

p231

www.ninjatrader.com
© NinjaTrader, LLC,
2005

El código de indicador es completo ahora y tiene que ser compilado. Usted puede compilar este indicador desde dentro el Redactor de NinjaScript menú de menú "Compile" de botón de ratón de derecho o simplemente presionar la llave F5. Es importante entender que este proceso hace el indicador listo para el uso de tiempo real y correrá natalmente dentro de NinjaTrader directamente. Esto no corre interpretado tan muchas otras aplicaciones hacen. Esto le provee del rendimiento más alto posible. Si hay cualquier error hizo un informe durante la compilación, los mensajes de error serán mostrados en el fondo del Redactor de NinjaScript.

-o-

Utilización

Su indicador está listo ahora para el uso y será puesto en una lista en la ventana Indicator Dialog.

p232

www.ninjatrader.co

m

© NinjaTrader, LLC, 2005

Una vez aplicado a una carta, puede mirar algo como la imagen abajo.

¡Felicitaciones, usted sólo completó la construcción de un indicador de encargo!

-0-

Seminario: Dibujo de Encargo (el Nivel 6)

p233

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Descripción

En este seminario de nivel 6 vamos a construir un indicador de encargo que es una variación CCI para mostrar opciones de dibujo diferentes para color de barra, color de línea y color de fondo.

-o-

Establecer

El primer paso en la creación de un indicador de encargo debe usar al mago de indicador de encargo. El mago generará el código de NinjaScript requerido que servirá de la fundación para su indicador de encargo.

1. Dentro de NinjaTrader Control Center ventana seleccionan los Instrumentos-> Nuevo NinjaScript-> Indicador... el menú 2. Presione el botón "Next>"

La definición de Propiedades de Indicador y Nombre Debajo de usted definirá su nombre de indicadores y varias propiedades de indicador.

3. Entre en la información como mostrado encima 4. Presione el botón "Next>"

p234

www.ninjatrader.co

m

© NinjaTrader, LLC, 2005

5. Este indicador requiere que un parámetro, Período, entre en la información como mostrado encima 6. Presione el botón "Next>"

La definición de Complots Debajo de usted definirá como su indicador es trazado en una carta.

7. Entre en la información como mostrado encima. Aunque el indicador CCI sólo tenga un complot, hemos definido tres complots encima. Esto es porque vamos a girar con./desc. la demostración de cada complot según el valor de CCI actual. Esto dará la ilusión que el indicador CCI

p235

www.ninjatrader.com
© NinjaTrader, LLC, 2005

la línea tiene tres colores diferentes. 8. Presione el botón "Next>"

Definición de Líneas de Oscilador

Debajo de usted definirá cualquier línea de oscilador requerida.

9. Entre en la información como mostrado encima. Añadimos dos líneas reference, un en +70 y otro en-70. 10. Apriete el "Siguiente>" el botón 11. Somos terminados ahora entrando en la información establecida de nuestro indicador. Presione el botón "Finish".

* en Cualquier momento, usted puede presionar el botón "Generate" en el mago si usted no tiene que pasar por cada uno de los pasos de mago

Usted verá ahora al Redactor de NinjaScript precargado por el código de NinjaScript generado por el mago. Debería mirar algo como la imagen abajo.

Su motivo de preocupación primario será los artículos "2" y "3".

1. Esta sección provee el indicador del nombre de demostración de carta y descripción usada en la ventana [Indicator Dialog](#). 2. Inicializar () la sección sólo es tratada una vez cuando el indicador es al principio cargado (añadido a una carta por ejemplo) y puede ser usado para establecer cualquier requisito de configuración de indicador. Usted puede ver que este mago generó el código que añadió el color de complot y diseñe esto definimos. 3. Esto es la sección que es llamada en cada actualización de barra (marca de entrada) y es donde usted entrará en su lógica de indicador

-o-

Entrada en Lógica de Cálculo

OnBarUpdate () el método es llamado para cada marca de entrada o en el final de una barra (usuario definido) realizando cálculos de tiempo real y es llamado en cada barra de una serie de datos calculando de nuevo el indicador. Por ejemplo, un indicador sería calculado de nuevo añadiéndolo a una carta existente que hace mostrar datos de precios existentes. Por lo tanto, esto es el método principal pidió el cálculo de indicador y usaremos este método de entrar en la escritura que calculará el valor de CCI y establecerá las condiciones usadas para poner el dibujo de encargo.

Inicializando el Indicador Inicializar () el método es llamado una vez antes de que cualesquiera datos de barra sean cargados y sean usados para configurar el indicador. El código abajo es automáticamente generado por el mago y añadido a Inicializar () método. Esto configura el indicador para tres complots y dos líneas y fija los parámetros.

p237

www.ninjatrader.com

Complots. Añada (nuevos Complots. Añada (nuevos Complots. Añada (nuevas Líneas. Añada (nuevas Líneas. Añada (nuevo Período = 14;
© NinjaTrader, LLC, 2005

```
Complot (Color. Magenta, PlotStyle. Línea, "Encima")); Complot  
(Color. Negro, PlotStyle. Línea, "Neutra")); Complot (Color.  
OrangeRed, PlotStyle. Línea, "Abajo")); Línea (Color. Negro, de  
70 años, "+70")); Línea (Color. Negro, -70, "-70"));
```

Entre en el código siguiente en Inicializar () el método y debajo del mago generó el código:

```
//Póngase la pluma solía dibujar el complot como unos Complots de línea rotos [2].Pen.  
DashStyle = DashStyle. Carrera;
```

El tercer complot llamado "Abajo" es de OrangeRed en color y tiene PlotStyle de la línea. El código encima cambia la propiedad de DashStyle de esta línea de una línea sólida a una línea rota. Esto es llevado a cabo teniendo acceso al complot vía **Complots [2]** sintaxis. Los nuevos complots siempre son añadidos a los **Complots** de colección en la secuencia. Basado en el mago generó el código encima, podemos ver que el complot llamado "Abajo" era el tercer complot añadido. Ya que las colecciones siempre son el cero basado pasamos en el valor de 2 (conspire 1 = 0, conspire 2 = 1, conspire 3 = 2 etc...) a entonces tienen acceso a Pen usada para dibujar el complot y cambiar la propiedad de DashStyle. Esto es un tema avanzado. El cambio de esta propiedad puede ser hecho por el usuario **del indicador** <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfssystemdrawingpenmemberstopic.asp> directamente dentro de la ventana Indicator Dialog. Usted puede leer más sobre la [clase de Pen aquí](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfssystemdrawingpenmemberstopic.asp).
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfssystemdrawingpenmemberstopic.asp>

Entre en el código siguiente en Inicializar () método y debajo del código entrado encima:

```
//Los umbrales de dibujo de línea establecidos que dan el efecto visual de//líneas  
coloreadas multi-basadas en Complots de valores de indicador [0].Min = 70; Complots  
[1].Max = 70; Complots [1].Min ==-70; Complots [2].Max ==-70;
```

El código encima pone los valores umbrales de complot. El concepto de poner valores umbrales debe ponerse cuando y cuando no pintar un complot en la carta. Este tema es cubierto en el mayor detalle en el [Seminario de Nivel 5](#).

El Dibujo de encargo Sustituye el código generado del mago por el código siguiente en OnBarUpdate () método en el Redactor de NinjaScript:

```
//No cuente si no tenemos bastantes barras si (CurrentBar <Período) vuelven;  
  
//Calcule el valor de CCI doble valor = CCI (Período) [0];  
  
//Use un si la rama para elegir un panel de indicador atrás basa el color y el color  
de barra si (valor> 200) {  
    BackColor = Color. PaleGreen; BarColor = Color. Amarillo;  
} más si (valor> 150) {  
    BackColor = Color. PaleGreen;} más si (valor <-200) {  
    BackColor = Color. Rosado; BarColor = Color. Amarillo;  
} más si (valor <-150) {  
    BackColor = Color. Rosado;}
```

El método primero comprueba para asegurar que hay bastantes barras para realizar el cálculo y luego adjudica el valor de CCI al **valor** variable. Un caso multi-si la declaración es usada para determinar cuando dibujar la gráfica de encargo.

```
si (valor> 200) {  
    BackColor = Color. PaleGreen; BarColor = Color. Amarillo;  
}
```

La primera rama comprueba si el valor es mayor que 200 y de ser verdadero, pone la propiedad de BackColor a PaleGreen y también pone la propiedad de BarColor al amarillo.

BackColor elige el color de fondo de la carta para la barra corriente. BarColor elige el color de la barra de precio actual.

Una lista de colores disponibles puede ser encontrada [aquí](#).

Cada uno sucesivo *más si* la rama comprueba el valor de valor y pone propiedades de dibujo de encargo.

El ajuste de los valores de complot Entra en el código siguiente en OnBarUpdate () método y debajo del código entrado encima:

```
//Ponga el valor de complot Encima. Juego (valor); Neutro. Juego (valor); Abajo. Juego (valor);
```

Para completar el indicador usted debe poner el valor de indicador a cada complot.

Inicializar () método en su redactor debería parecer idéntico a la imagen abajo:

m
© NinjaTrader, LLC, 2005

OnBarUpdate () método en su redactor debería parecer idéntico a la imagen abajo:

-o-

p240

www.ninjatrader.com
© NinjaTrader, LLC,

2005

Compilar

El código de indicador es completo ahora y tiene que ser compilado. Usted puede compilar este indicador desde dentro el Redactor de NinjaScript menú de menú "Compile" de botón de ratón de derecho o simplemente presionar la llave F5. Es importante entender que este proceso hace el indicador listo para el uso de tiempo real y correrá natalmente dentro de NinjaTrader directamente. Esto no corre interpretado tan muchas otras aplicaciones hacen. Esto le provee del rendimiento más alto posible. Si hay cualquier error hizo un informe durante la compilación, los mensajes de error serán mostrados en el fondo del Redactor de NinjaScript.

-o-

Utilización

Su indicador está listo ahora para el uso y será puesto en una lista en la ventana Indicator Dialog.

Una vez aplicado a una carta, puede mirar algo como la imagen
abajo.

p241

www.ninjatrader.co

m

© NinjaTrader, LLC,
2005

¡Felicitaciones, usted sólo completó la construcción de un indicador de encargo!

-o-

Desarrollo de Estrategias de Encargo

Descripción

NinjaScript permite que usted desarrolle estrategias de encargo en el tiempo rápido usando más de 100 indicadores de sistema incorporados, 3ros indicadores del partido o [indicadores de encargo](#). Las estrategias NinjaScript son compiladas y dirigidas natalmente dentro de la aplicación de NinjaTrader que proporciona el rendimiento más alto posible.

Por favor lleve tiempo para examinar esta sección incluso el [Proceso de desarrollo de Estrategia](#).

Descripciones tutelares Todos los indicadores NinjaTrader internos y las estrategias de muestra vienen con el código fuente lleno y pueden ser vistos dentro del Redactor de NinjaScript. Por favor examine los seminarios en esta sección para el paseo detallado throughs del desarrollo de estrategia de encargo.

[El nivel 1](#) - Transición de MAMÁ Simple (Demuestra la construcción de estrategia por el mago y scripting) [el Nivel 2](#) - RSI con el Objetivo de Ganancia & Pérdida de Parada (scripting sólo)

-o-

El Proceso de desarrollo de Estrategia

Describe su Estrategia que Describe sus medios de estrategia de crear un juego de reglas objetivas que definen las condiciones usadas para entrar y salir de un mercado. La descripción de su estrategia siempre comienza con la Estrategia

p242

www.ninjatrader.com

© NinjaTrader, LLC, 2005

El mago y luego proporciona las opciones siguientes:

- [Mago de estrategia con el Constructor de Condición](#) - Esto es un punto y enfoque de chasquido para la estrategia la descripción que es ideal para cada uno del no programador, programador de principiante y programador avanzado.
- [Redactor de NinjaScript](#) - Esto es un redactor scripting moderno con la comprobación de sintaxis inline llena y intellisense. Esto es un gran enfoque para aquellos que quieren cifrar a mano su lógica de estrategia. Si usted va a mí cifran su estrategia, por favor son familiares con [Inicializar \(\)](#)

y [OnBarUpdate \(\)](#) métodos.

Backtest y Optimize su Estrategia Una vez que usted ha completado la descripción de su estrategia que usted puede probarlo entonces contra datos históricos para determinar objetivamente como la estrategia funcionó en un mercado (s) específico en el pasado.

- [Estrategia Analizador](#) - Usted puede backtest, optimizar, y analizar sus resultados históricos

A este punto en el proceso usted pasará probablemente por un ciclo iterativo por donde usted cambia su descripción de estrategia, backtest, descripción de cambio y backtest hasta que usted tenga una estrategia que cumple con sus requisitos.

Ver instrucciones especiales en [estrategias FX que corren](#).

La Prueba de tiempo real su Estrategia es crítico que antes de que usted despliegue su estrategia contra su cuenta de mercancías viva que usted lo prueba en la operación de tiempo real para asegurar que la mecánica (la operación) de su estrategia se comporta ya que usted lo esperaría a. Además, usted también puede expedir a prueba su estrategia usando datos de mercado de tiempo real contra el [motor de simulación de comercio](#) de NinjaTrader. NinjaTrader proporciona varias opciones para pruebas de tiempo real:

- [Conexión de Comida de Datos simulada](#) - Esto es un mercado arbitrario internamente generado con el usuario

la tendencia controlada y es grande para la operación de pruebas de fuerza de una estrategia • [la Conexión de Repetición de Mercado](#) - Registro, la repetición en el usuario definió velocidades mercados múltiples

simultáneamente y dirigido sus estrategias • [Simulación de Tiempo real](#) - Se unen con su agente de bolsa o vendedor de datos de mercado en de tiempo real y dirigido sus estrategias por nuestro motor de simulación de tecnología avanzada

Usted puede [dirigir su estrategia](#) de una carta o de la etiqueta de Estrategias de la ventana Control Center. Usted puede generar datos [de rendimiento de estrategia](#) de tiempo real de la etiqueta de Estrategias.

Corriendo en su Cuenta de mercancías Viva Ahora que usted ha descrito, backtested y de tiempo real probó su estrategia, usted está listo para [automatizar su estrategia contra su cuenta de mercancías viva](#). Unas puntas usted debería saber:

- Por favor asegúrese que usted totalmente entiende que las [opciones de tiempo de ejecución vivas](#) • [rendimiento](#) de estrategia Live [variarán](#) de sus resultados de backtested

- Por favor asegúrese que usted totalmente entiende [la Posición de Estrategia contra la Posición de Cuenta... su](#)

la posición de estrategia no es un a una relación con su posición de cuenta de corretaje... usted tendría que sincronizar si ellos deberían ser sincronizados

- Las estrategias son automáticamente terminadas (deje de correr) en • cerrado de NinjaTrader el comercio Automatizado no significa van pescando mientras su ordenador comercia para usted, nosotros muy

recomiende que usted sea dentro de la proximidad inmediata con su ordenador mientras esto corre y estrategia comercial automatizada, usted nunca sabe lo que puede

- equivocarse
- Usted puede dirigir estrategias comerciales múltiples al mismo tiempo en el mismo mercado

p243

www.ninjatrader.com
© NinjaTrader, LLC, 2005

-o-

Posición de estrategia Contra Posición de Cuenta

Un concepto importante para entender antes de la utilización de estrategias NinjaScript en un ambiente comercial de tiempo real (la cuenta de corretaje viva por ejemplo) es la diferencia entre una *Posición de Estrategia* y una *Posición de Cuenta*. Una *Posición de Estrategia* es una posición virtual que es creada por la entrada y ejecuciones de salida generadas por una estrategia y es independiente de la cualquier otra posición de estrategia de marcha o una posición de cuenta. Una *Posición de Cuenta* es la posición que usted realmente sostiene en una cuenta de mercancías de tiempo real, si es NinjaTrader cuenta de simulación interna (Sim101) o su cuenta de corretaje de verdadero dinero viva. En mayoría de los casos, un comerciante querría que su talla de *Posición de Estrategia* y dirección de mercado sean iguales (en la sincronización) a su *Posición de Cuenta* pero hay situaciones cuando esto puede no ser el caso.

Por ejemplo:

- Usted quiere dirigir estrategias múltiples en el mismo mercado simultáneamente donde estrategia A sostiene una 1 posición LARGA, la estrategia B sostiene una 2 posición LARGA que causa una cuenta que debería sostener una 3 posición LARGA a fin de estar en la sincronización con ambas estrategias
- Usted quiere dirigir una estrategia y al mismo tiempo cambiar el mismo mercado que la estrategia es la marcha en utilización de táctica discrecional por una de la ventana de entrada de pedido avanzada de NinjaTrader, como el Comerciante de Carta o SuperDOM

Un escenario... muy común

Un guión muy común comienza una estrategia NinjaScript en medio de una sesión comercial, tal como una hora después de que la sesión ha comenzado.. La estrategia NinjaScript es dirigida en cada barra histórica para la 1ra hora de la sesión (esto realmente correrá en todos los datos históricos cargados en una carta) decidir que la situación actual declara que estaría en si esto hubiera estado corriendo vivo desde el principio de la sesión. Este estado de posición entonces se hace la *Posición de Estrategia* para su estrategia. Déjenos suponer que durante la hora histórica su estrategia hubiera entrado en una 1 posición LARGA y la posición todavía está abierta. Esto significaría que la *Posición de Estrategia* es LARGA 1 y ya que este comercio realmente no fue ejecutado en una cuenta su *Posición de Cuenta* es LLANA.

¿Qué puede usted hacer en este caso?

Si usted quiere que su *Posición de Cuenta* corresponda a su *Posición de Estrategia*, usted tendrá que colocar un pedido manual en la cuenta en la cual la estrategia corre. Siguiendo

del susodicho ejemplo, usted tendría que colocar un orden al precio de mercado de 1 parte para el mercado cambiado en la cuenta en la cual la estrategia corre.

¿Y si yo no presente un pedido manual de sincronizar mi cuenta?

El comportamiento que resulta cuando la *Posición de Cuenta* y *Posición de Estrategia* son fuera de la sincronización consiste en cuando su estrategia (siguiendo con el ejemplo encima) cierra la posición larga con una orden de venta esto traería la *Posición de Estrategia* al piso y su *Posición de Cuenta* a 1 CORTO.

-O-

Multimarca de tiempo & Instrumentos

Una estrategia NinjaScript apoya multimarca de tiempo e instrumentos en una estrategia sola. Esto es posible ya que usted puede añadir que las Barras adicionales se oponen a una estrategia. Un objeto de Barras representa todas las barras de datos de una carta. Si usted tuviera un MSFT carta de 1 minuto con barras de 200 minutos en ello, las barras de 200 minutos representan un objeto de Barras. Usted puede ejecutar hasta comercios a través de todos los instrumentos diferentes en una estrategia. Hay flexibilidad extrema en el modelo de estrategia que NinjaTrader usa para estrategias de barras múltiples entonces es muy importante que antes de que usted incorpore objetos de Barras adicionales a una estrategia, usted entienda como todo esto trabaja. Un hecho importante para entender es que una estrategia NinjaScript es realmente el acontecimiento conducido; cada Barras

p244

www.ninjatrader.com
© NinjaTrader, LLC, 2005

el objeto en una estrategia llamará OnBarUpdate () método. ¿Qué significa esto? Se hará evidente ya que usted lee adelante.

También es importante que usted entienda el método siguiente y propiedades:

• [Añada \(\)](#) • [BarsArray](#) • [BarsInProgress](#)

El entendimiento de esta Sección (DEBE LEER) Ya que nos movemos por esta sección, el término "primarias de Barras" será usado y para la aclaración, esto siempre se referirá al primer objeto de Barras cargado en una estrategia. Por ejemplo, si usted aplica una estrategia de MSFT carta de 1 minuto, las Barras primarias serían MSFT conjunto de datos de 1 minuto.

Esta sección es escrita de la moda secuencial. El ejemplo de código es reutilizado y construido en de la sección de suscripción a la sección de suscripción.

Añadiendo el Objeto de Barras Adicional a una Estrategia las Barras Adicionales son añadidas a una estrategia vía Añadir () método en [Inicializar \(\)](#) método. Cuando un objeto de Barras es añadido a una estrategia, también es añadido a la serie de BarsArray. ¿Qué es que usted pregunta? Piense en ello como un contenedor en la estrategia que sostiene todos los objetos de Barras añadidos a la estrategia. Es no diferente que un balde de bolas de golf en su garaje. Como un objeto de Barras es añadido a la estrategia, él es añadido a este contenedor, BarsArray, y dado un número índice entonces podemos recuperar este las

Barras se oponen más tarde. No sude esto si parece complejo, su completamente fácil una vez que usted lo ve en el uso práctico. Su explicado en mayores detalles más tarde en esta sección.

Para la demostración, vaya a suponer que un MSFT barra de 1 minuto es nuestras Barras primarias (juego cuando la estrategia es aplicada a un 1 minuto carta de MSFT) y que Inicializar () método añade un objeto de Barras de 3 minutos de MSFT y luego añade un objeto de Barras de un 1 minuto de AAPL para un total de 3 objetos de Barras únicos.

```
protegido anulan el vacío Inicializan () {  
    Añada (PeriodType. Minuto, 3); Añada ("AAPL", PeriodType. Minuto, 1);  
}
```

Como los Datos de Barra son Referidos Entendiendo como las barras de multimarca de tiempo son tratadas y que datos OHLCV son referidos es crítico.

La imagen "de Figura 1" abajo demuestra el concepto del procesamiento de barra en datos históricos o en el de tiempo real cuando la propiedad [de CalculateOnBarClose](#) es puesta al verdadero. Las barras de 1 minuto en el amarillo sólo sabrán el OHLCV de la barra de 3 minutos en el amarillo. Las barras de 1 minuto en el cian sólo sabrán los datos OHLCV de la barra de 3 minutos en el cian. Mire a "la Barra 5" que es la barra de minuto de 5ta, si usted quisiera saber el valor alto corriente para durante la marca de tiempo de 3 minutos, usted conseguiría el valor de la 1ra barra de 3 minutos ya que esto es la última barra "cerrada". La 2da barra de 3 minutos (cian) no es conocida en este tiempo.

p245

www.ninjatrader.co

m

© NinjaTrader, LLC,
2005

Contrastan la susodicha imagen y concepto con la imagen debajo de la cual demuestra el procesamiento de barra en el de tiempo real cuando la propiedad de CalculateOnBarClose es puesta al falso (marca por el procesamiento de marca). Las barras de 1 minuto en el

amarillo sabrán OHLCV corriente de la barra de 3 minutos en el amarillo (2da barra de 3 minutos) que todavía está en la formación... todavía no cerrada.

El punto es si usted tiene una estrategia de multimarca de tiempo en el de tiempo real y trata la marca por la marca en vez de en el final de cada barra, entienda que los datos OHLCV a los cuales usted tiene acceso en el de tiempo real son diferentes que en datos históricos.

Un ejemplo rápido para ilustrar el punto:

Su estrategia tiene la lógica compleja que cambia el color de barra en la carta. Usted dirige la marca por la marca, según la susodicha imagen "de Figura 2", la barra de 1 5to minuto mira datos OHLCV de la 2da barra de 3 minutos. Su estrategia cambia el color de barra de 1 5to minuto al verde. En el futuro usted recarga su estrategia en la carta (por cualesquiera motivos) y la barra de 1 5to minuto es ahora una barra histórica. Según la imagen "de Figura 1", la barra de 1 5to minuto ahora se refiere a los datos OHLCV de la 1ra barra de 3 minutos (en vez de la 2da barra de 3 minutos según la Figura 2) y como consiguiente, su condición de lógica de estrategia para colorear la barra verde está caducada. Ahora su carta parece diferente.

Usando Objetos de Barras como la Entrada a Métodos de Indicador En la sección de suscripción encima, el concepto de valores de índice fue introducido. Esto es un concepto crítico para entender ya que es usado consecuentemente trabajando con estrategias de multibarras.

Vaya a demostrar este concepto:

Continuando del ejemplo encima, nuestras Barras primarias es puesto de un MSFT carta de 1 minuto

Dan a MSFT Barras de 1 minuto un valor de índice de 0

En Inicializar () método añadimos un objeto de Barras de 3 minutos MSFT y un AAPL que Barras de 1 minuto objetan a la estrategia

Dan a MSFT Barras de 3 minutos un valor de índice de 1

Dan a AAPL Barras de 1 minuto un valor de índice de 2

Dan valores de índice a cada objeto de Barras ya que ellos son añadidos a una estrategia de una moda incremental. Si hay 10 objetos de Barras en una estrategia, entonces usted tendrá valores de índice en los límites de 0 a 9.

Nuestra estrategia ahora tiene 3 objetos de Barras en BarsArray de contenedor. De este punto adelante, podemos pedir a este contenedor darnos las Barras objetan a que queramos trabajar con proporcionando el valor de índice. La sintaxis para esto es:

BarsArray [índice]

Esto permite que nosotros consigamos el objeto de Barras correcto y lo usemos como introducido para un [método de indicador](#). Por ejemplo:

ADX (14) [0]> 30 && ADX (BarsArray [2], 14) [0]> 30

La susodicha expresión en inglés traduciría a:

Si el 14 período ADX de MSFT 1 minuto es mayor que 30 y el 14 período ADX de APPL 1 minuto es mayor que 30

El ejemplo siguiente comprueba si el valor de CCI corriente para todos los objetos de Barras está encima 200. Usted notará que BarsInProgress es usado. Esto debe comprobar qué objeto de Barras llama OnBarUpdate () método. Más en esto más tarde en esta sección.

```
protegido anulan OnBarUpdate vacío () {
    si (BarsInProgress == 0) {
        si (CCI (20) [0]> 200 && CCI (BarsArray [1], 20) [0]> 200
            && CCI (BarsArray [2], 20) [0]> 200)
        {
            //Haga algo
        }
    }
}
```

Acontecimiento verdadero Conducido en OnBarUpdate () el Método Desde NinjaScript es realmente el acontecimiento conducido, OnBarUpdate () el método es llamado para cada acontecimiento de actualización de barra para cada objeto de Barras añadido a una estrategia. Este modelo proporciona la flexibilidad suma. Por ejemplo, usted podría hacer combinar sistemas de comercio múltiples en un dependiente de estrategia el uno en el otro. Expresamente, usted podría tener un 1 minuto objeto de Barras de MSFT y un 1 minuto objeto de Barras de AAPL, tratar reglas comerciales diferentes de cada objeto de Barras y comprobar para ver si MSFT es largo cuando AAPL lógica comercial está siendo tratado.

La propiedad [de BarsInProgress es](#) usada para identificarse qué objeto de Barras llama OnBarUpdate () método. Esto permite que usted elimine los acontecimientos a los cuales usted quiere o no quiere tratar.

Siguiendo nuestro ejemplo encima, vaya a mirar a algún código para entender mejor lo que pasa. Recuerde, tenemos tres objetos de Barras que trabajan en nuestra estrategia, unas Barras primarias MSFT 1 minuto, MSFT de 3 minutos y AAPL 1 minuto.


```

protegido anulan OnBarUpdate vacío () {
    //Los controles si OnBarUpdate () es llamado de una actualización en las Barras
    primarias si (BarsInProgress == 0) {
        si (Cerca [0]> Abierto [0])
            //Haga algo}

    //Los controles si OnBarUpdate () es llamado de una actualización en Barras de
    3 minutos MSFT si (BarsInProgress == 1) {
        si (Cerca [0]> Abierto [0])
            //Haga algo}

    //Los controles si OnBarUpdate () es llamado de una actualización en AAPL Barras
    de 1 minuto si (BarsInProgress == 2) {
        si (Cerca [0]> Abierto [0])
            //Haga algo
    }}

```

Lo que es importante para entender en el susodicho código de muestra es que tenemos "**si**" las ramas que comprueban para ver lo que Excluye el objeto llaman OnBarUpdate () método a fin de tratar la lógica comercial relevante. Si sólo quisiéramos tratar los acontecimientos de las Barras primarias podríamos escribir nuestra primera declaración como sigue:

```

si (iBarsInProgress! = 0)
vuelva;

```

Lo que también es importante para entender es el concepto del contexto. Cuando OnBarUpdate () el método es llamado, será llamado dentro del contexto del objeto de Barras de vocación. Esto significa que si las Barras primarias provoca OnBarUpdate () el método, todos los métodos de indicador y los datos de precios señalarán a los aquellos datos de objeto de Barras. Mirando el susodicho ejemplo, ver cómo la declaración "**si (¿Cerca [0]> Abierto [0])**" existe bajo cada uno "**si**" rama? Los valores devueltos por el Cercano [0] y Abierto [0] serán los valores de precios cercanos y abiertos para el objeto de Barras de vocación. Entonces cuando BarsInProgress == 0 (Barras primarias) el valor cercano volvió es el precio cercano del MSFT barra de 1 minuto. Cuando BarsInProgress == 1 el valor cercano volvió es el precio cercano del objeto de Barras de 3 minutos MSFT.

Tener acceso a los Datos de Precios en una Estrategia de Multibarras Como usted probablemente sabe ya, usted puede tener acceso a la cotización de cierre de la barra corriente con la declaración siguiente:

Cerca [0];

Usted también puede tener acceso a datos de precios, como el precio cercano de otros objetos de Barras en cualquier momento. Esto es llevado a cabo teniendo acceso a [Abrir, Máximos, Mugidos, Finales, Volúmenes, Medianas, Typicals](#) y serie [de Times](#) por el valor de índice. Estas propiedades sostienen colecciones (contenedores) que sostienen sus valores llamados para todos los objetos de Barras en una estrategia.

Seguir con nuestro ejemplo de código encima, si usted quisiera tener acceso al alto precio del MSFT 3 objeto de Barras de minuto que está en el índice 1 que usted escribiría:

Máximos [1] [0];

Esto dice sólo me dan la serie de altos precios por el objeto de Barras en el índice 1 "**Máximos [1]**" y devuelven a mí el valor alto corriente "**[0]**". Ahora, si el índice de BarsInProgress era igual a

p248

www.ninjatrader.com

© NinjaTrader, LLC, 2005

1, el contexto corriente es del MSFT 3 objeto de Barras de minuto entonces usted podría escribir sólo:

Alto [0];

El ejemplo siguiente demuestra varios modos de tener acceso a datos de precios.

```
protegido anulan OnBarUpdate vacío () {
    //Los controles si OnBarUpdate () es llamado de una actualización en las
    Barras primarias si (BarsInProgress == 0) {
        doble primaryClose = Cerca [0]; doble msft3minClose = Finales [1] [0];
        doble aapl1minClose = Finales [2] [0];

        //el primaryClose también podría ser expresado como//primaryClose =
        Finales [0] [0];
    }

    //Controles si OnBarUpdate () es llamado de una actualización en Barras de
    3 minutos MSFT
obj
eto
    si (BarsInProgress == 1) {
        dóblese primaryClose = se doblan msft3minClose = doblan aapl1minClose =
    }

    Finales [0] [0]; Cerca [0]; Finales [2] [0];
}
```

La entrada y la Salida y Recuperar métodos de Salida y Entrada **de información de Posición** son ejecutados dentro del contexto de BarsInProgress. Vaya a manifestarse con un ejemplo:

```
protegido anulan OnBarUpdate vacio () {  
    //Los controles si OnBarUpdate () es llamado de una actualización en las Barras  
    primarias si (BarsInProgress == 0) {  
        //Presenta un orden al precio de mercado de compra para EnterLong MSFT ();  
    }  
  
    //Los controles si OnBarUpdate () es llamado de una actualización en Barras de  
    3 minutos MSFT  
objeto  
    si (BarsInProgress == 1) {  
        //Presenta un orden al precio de mercado de compra para EnterLong MSFT ();  
    }  
  
    //Los controles si OnBarUpdate () es llamado de una actualización en AAPL Barras  
    de 1 minuto  
objeto  
    si (BarsInProgress == 2) {  
        //Presenta un orden al precio de mercado de compra para EnterLong AAPL ();  
  
        //Presenta un mercado de compra para MSFT cuando OnBarUpdate () es llamado  
        EnterLong AAPL (0, 100, "COMPRE MSFT");  
    }  
}
```

Como usted puede ver encima, los pedidos son presentados para MSFT cuando

BarsInProgress es igual a 0 y para AAPL cuando BarsInProgress es igual a 2. Los pedidos presentados son dentro del contexto del objeto de Barras que llama OnBarUpdate () método y el instrumento asociado al objeto de Barras de vocación. Hay una excepción que es el pedido colocado para MSFT dentro del contexto de OnBarUpdate () llamada a APPL. Cada método de pedido tiene una variación que permite que usted especifique el valor de índice de BarsInProgress que permite la sumisión de pedidos por cualquier instrumento dentro del contexto de otro instrumento.

La Posición de propiedad siempre se refiere a la posición del instrumento del contexto corriente. Si BarsInProgress es igual a 2 (APPL Barras de 1 minuto), la Posición se referiría a la posición sostenida para APPL. Las [Posiciones](#) de propiedad sostienen una colección de objetos de Posición para cada **instrumento** en una estrategia. Note que hay una diferencia crítica aquí. En todas partes de esta sección entera hemos estado tratando con objetos de Barras. Aunque en nuestra muestra tengamos tres objetos de Barras (MSFT 1 y 3 minutos y APPL 1 minuto) sólo tenemos dos instrumentos en la estrategia.

Dan la posición de MSFT dan un valor de índice de 0 posición AAPL un valor de índice de 1

En el ejemplo abajo, cuando OnBarUpdate () el método es llamado para las Barras primarias también comprobamos si la posición sostenida para AAPL no es llana y luego entre en una posición larga en MSFT. El resultado neto de esta estrategia consiste en que participan en una posición larga APPL, y luego una vez que APPL es largo, vamos mucho tiempo MSFT.

p250

www.ninjatrader.co

m

© NinjaTrader, LLC,
2005

```
protegido anulan OnBarUpdate vacio () {
    //Los controles si OnBarUpdate () es llamado de una actualización en las Barras
    primarias si (;BarsInProgress == 0 && Posiciones [1].MarketPosition! =
MarketPosition. Piso)
    {
        //Presenta un orden al precio de mercado de compra para EnterLong MSFT ();
    }

    //Los controles si OnBarUpdate () es llamado de una actualización en AAPL Barras
    de 1 minuto si (BarsInProgress == 2) {
        //Presenta un orden al precio de mercado de compra para EnterLong APPL ();
    }
}
```

Utilización de Estrategias de ATM

Usted puede crear una estrategia automatizada que genera una señal comercial que ejecuta una [Estrategia de ATM](#) de NinjaTrader.

- Las Estrategias de ATM funcionan en el de tiempo real sólo y no ejecutarán en datos históricos así ellos
no puede estar de vuelta Ejecuciones • probadas que resultan de una Estrategia de ATM que es creada desde dentro NinjaScript
la estrategia automatizada no conspirará en una carta durante la operación de tiempo real
- la Estrategia estableció parámetros, como [EntriesPerDirection](#), [EntryHandling](#), [ExitOnClose](#) no hacen
preséntese llamando [AtmStrategyCreate \(\)](#) método

Hay una Línea Clara... Hay una línea clara entre una Estrategia de NinjaScript y una Estrategia de ATM. El modelo de uso para crear una Estrategia de ATM dentro de una Estrategia de NinjaScript es cuando usted quiere a programatically supervisan y generan una señal de entrada y luego los manualy manejan el puesto vacante que resulta vía una Estrategia de ATM en una de [las ventanas de entrada de pedido](#) de NinjaTrader.

!!!! IMPORTANTE: A mano Cerrando una Estrategia de ATM de una Ventana de Entrada de Pedido, como SuperDOM es crucial que dirigiendo Estrategias de ATM creadas por una estrategia NinjaScript a la cual usted entiende como correctamente a mano cerca la Estrategia de ATM de cualquiera de las ventanas de entrada de pedido.

- Si el Modo [de Selección de Estrategia de ATM](#) de ventana de entrada de pedido no está en
"DisplaySelectedATMStrategyOnly" hacen clic en el botón "CLOSE" vía su botón de ratón medio (rueda de voluta)
- Si el Modo de Selección de Estrategia de ATM de ventana de entrada de pedido está en "DisplaySelectedATMStrategyOnly" usted puede hacer clic en el botón "CLOSE" con su botón de ratón izquierdo para cerrar la estrategia de ATM activa seleccionada

Después de enfoques encima de la voluntad los internallly cierran la Estrategia de ATM. No después de enfoque cerrará la posición de cuenta/instrumento, terminará todas las estrategias y anulará todos los pedidos. El resultado consiste en que su estrategia NinjaScript será terminada.

Puntas y Muestras de Referencia

Hay un depósito creciente de [puntas](#) de NinjaScript y muestras de referencia en <http://www.ninjatrader-support.com/vb/forumdisplay.php?f=29> > el Foro de Apoyo de

NinjaTrader.

-o-

Las mejores Prácticas para 3ros Reveladores del Partido

Por favor ver la sección ["las Mejores Prácticas"](#).

-o-

La utilización de 3ros Indicadores del Partido

Usted puede usar 3ros indicadores del partido dentro de sus estrategias o indicadores de encargo. Un 3er indicador del partido es un indicador que no fue desarrollado por NinjaTrader. Es **importante entender del 3er revelador del partido, la funcionalidad proporcionada o **NO** proporcionado en sus indicadores patentados. Sólo porque ellos proporcionan un indicador que muestra una tendencia optimista o pesimista en una carta **no significa** que usted puede tener acceso a este estado de tendencia de su indicador.**

Los 3ros indicadores del partido pueden ser proporcionados a usted de uno de los modos siguientes:

- NinjaScript archivan el archivo que puede ser directamente [importado](#) en NinjaTrader
- Un instalador de encargo
- Un juego de archivos e instrucciones para salvarlos en las carpetas correctas

Si usted fuera proveído de un archivo de archivo de NinjaScript que usted ha importado con éxito vía el menú "File> Utilities> Import NinjaScript" de ventana Control Center que usted es puede saltarse sobre la información abajo ya que NinjaTrader automáticamente configura los indicadores listos para el uso.

Si usted fuera proveído de un instalador de encargo o una asamblea compilada (.DLL) archivo que usted tuvo que salvar a mano en la carpeta Mis Documentos \<Carpeta de NinjaTrader> \bin\Custom entonces usted debe seguir las instrucciones abajo.

El vendedor Archiva al 3er revelador del partido debería haber instalado o un archivo "de Vendedor" o haberle proveído de uno. Su probable en el formato "NinjaTrader.VendorName.cs" donde VendorName es el nombre del 3er vendedor del partido. Este archivo permite que usted tenga acceso cómodamente a sus indicadores.

- Si le proveyeran un instalador, usted puede concordar con el vendedor si este archivo fuera incluido o; los
- Si ellos le proveyeran este archivo, lo salvan a "Mis Documentos \<Carpeta de NinjaTrader> \bin\Custom" y reanude NinjaTrader

La adición de una Referencia 1. Desde dentro cualquier indicador NinjaScript o estrategia (los Instrumentos> Corrigen NinjaScript> Indicador y abren cualquier indicador en el Redactor de NinjaScript), el chasquido correcto en su ratón para subir el menú de contexto y seleccionar el submenú "References..." según la imagen abajo

p252

www.ninjatrader.co

m

© NinjaTrader, LLC, 2005

2. Una ventana "References" aparecerá 3. Presione el botón "Add" y seleccione al 3er vendedor del partido archivo de DLL

4. Usted verá una referencia al 3er vendedor del partido DLL en la ventana 5 "de Referencias". Presione el botón "OK"

Usted será capaz ahora de tener acceso a los métodos de indicador proporcionados por el 3er vendedor del partido.

-o-

Seminario: Transición de MAMÁ simple (el Nivel 1)

Descripción

En este seminario de nivel 1 vamos a construir la estrategia automatizada de una costumbre que va mucho tiempo

p253

www.ninjatrader.com
© NinjaTrader, LLC, 2005

cuando el valor medio móvil rápido se cruza encima del valor medio móvil lento y va corto
cuando el valor medio móvil rápido se cruza debajo del valor medio móvil lento.

-o-

Establecer

El primer paso en la creación de una estrategia de encargo debe usar al mago de estrategia de encargo. El mago proporciona dos opciones:

- Permita usted para crear una estrategia funcional sin cualquier programación •

Genera el código de NinjaScript requerido que servirá de la fundación para su costumbre estrategia para codificación adicional

1. Dentro de NinjaTrader Control Center ventana seleccionan los Instrumentos> Nuevo NinjaScript> Estrategia... el menú 2. Presione el botón "Next>"

3. Entre en la información como mostrado encima 4. Presione el botón "Next>"

La definición de Parámetros de Entrada Debajo de usted definirá sus parámetros de entrada de estrategias. Éstos son cualquier parámetro de entrada que pueda ser cambiado por el usuario corriendo o backtesting una estrategia. Si su estrategia no requiere que cualquier parámetro deje los campos "de Nombre" en blanco.

p254

www.ninjatrader.co

m

© NinjaTrader, LLC,
2005

5. Añada las entradas según la imagen encima 6. Presione el botón "Next>"

La definición de Condiciones y Acciones Debajo de usted puede definir condiciones que provocan las acciones definidas del usuario, como colocación de pedidos, usar una carta o creación de una alarma.

Note como hay dos botones en la pantalla abajo:

El Código de visión... - Presionando este botón carga el código de estrategia en el Redactor de NinjaScript para ver objetivos sólo. Esto es un gran enfoque si usted es nuevo a la programación o usted quiere ver como el mago de estrategia dinámicamente genera el código de escritura correcto en marcha.

Abra el Código - Presionando estas cargas de botón el código de estrategia en el redactor de NinjaScript para la edición manual adicional. Una vez que este botón es presionado, usted no puede volver al Mago para construcción de estrategia y edición.

Si usted quiere seguir con este seminario por [mí la programación sigue aquí](#) después de presionar el botón "Unlock Code".

-o-

La creación de Strategy vía el Mago

1. Presione el botón "Add" para mostrar la ventana "Condition Builder" según la imagen abajo

2. Amplíe la sección "de Indicadores" haciendo clic en "el +" botón y destaque el indicador "SMA"

3. Seleccione la propiedad "de Período" en la ventana izquierda inferior y apriete "el..." botón para abrir la ventana "Value"

4. De la ventana "Value", amplíese el "Usuario definió entradas" sección y seleccione la entrada "Rápida"

5. Prensa "OKEY" para adjudicar al usuario define la entrada "Rápido" a la propiedad "de Período" SMA 6. Permita a este indicador ser trazado en una carta

7. Seleccione "CrossAbove" y ponga el período de mirada hacia atrás a un valor de "1" 8. Seleccione el indicador "SMA" en la ventana 9 correcta. Póngase el período "Lento" (justo como usted hizo para Rápido en el paso 4) 10. Permita a este indicador ser trazado en una carta

p257

www.ninjatrader.co

m

© NinjaTrader, LLC, 2005

Si usted mira la imagen encima, usted sólo creó una condición inicial. La condición es "si el

valor medio móvil simple rápido se cruza encima del valor medio móvil simple lento".

11. Presione el botón "OK" y usted verá la imagen siguiente

12. Presione el botón "Add" en la sección correcta inferior del Mago para subir la "ventana 13 de diálogo" de Acción de Estrategia. Amplíe la "Sección" de dirección de pedido haciendo clic en "el +" botón y destaque el "Incorporan la posición Larga" acción y luego presionan el botón "OK"

14. Seleccione la condición según la imagen abajo, chasquido correcto y seleccione "la Copia" del menú 15. Haga clic en el "Juego en 2" etiqueta y mueva su cursor en el área de condiciones, chasquido correcto y seleccione "la Pasta" del menú

16. Apriete "el..." botón para la condición copiada que criará al Constructor de Condición 17. En el Constructor de Condición, cambie la función "de CrossAbove" a "CrossBelow" 18. También queremos poner "al falso" el "Complot en la carta" parámetros (Paso #6) ya que ya los ponemos al verdadero en "el Juego 1" etiqueta 19. Apriete "el OKEY" el botón 20. Repita pasos #12 y #13 la excepción substituting "Entra posición corta" para "Entran en la posición larga"

p259

www.ninjatrader.com
© NinjaTrader, LLC, 2005

21. Complete el resto del mago apretando el "Siguiente>" buttton

-o-

La creación de la Estrategia vía Mí Programación

Si usted no ha hecho tan ya, presione el botón "Unlock Code" dentro del mago para lanzar al [Redactor de NinjaScript](#).

OnBarUpdate () método es llamado para cada marca de entrada o en el final de una barra (usuario definido) realizando cálculos de tiempo real. Por lo tanto, esto es el método principal pidió el cálculo de estrategia y usaremos este método de entrar en la escritura que comprueban condiciones de salida y entrada.

La Condición de Salida y Entrada Entra en el código contenido dentro de OnBarUpdate () método a la imagen abajo en OnBarUpdate () método en el Redactor de NinjaScript:

Traducido a inglés, el código contenido dentro de OnBarUpdate () el método encima lee:

*si el valor medio móvil simple rápido se cruza **encima del** valor medio móvil simple lento dentro de la última barra, vaya mucho tiempo*

*si el valor medio móvil simple rápido se cruza **debajo del** valor medio móvil simple lento dentro de la última barra, vaya corto*

Para llevar a cabo esto usamos los métodos siguientes:

[CrossAbove \(\)](#) - Controles de una cruz encima de condición y vueltas [CrossBelow](#) verdadero o falso [\(\)](#) - Controles de una cruz debajo de condición y vueltas [SMA](#) verdadero o falso [\(\)](#) - Vueltas el valor de un valor medio móvil simple [EnterLong \(\)](#) - Entra en un orden al precio de mercado [EnterShort](#) largo [\(\)](#) - Entra en un orden al precio de mercado corto

o-

Compilación

El código de estrategia es completo ahora y tiene que ser compilado.

- Si usted completara este seminario vía el Mago de Estrategia, simplemente siga las instrucciones de mago al final a cual tiempo la estrategia compilará.
 - Si usted mí cifrara este seminario usted puede compilar esta estrategia desde dentro NinjaScript
- El menú de menú "Compile" de botón de ratón de derecho de redactor o simplemente presiona la llave F5.

Es importante entender que este proceso hace la estrategia lista para el uso de tiempo real y correrá natalmente dentro de NinjaTrader directamente. Esto no corre interpretado tan muchas otras aplicaciones hacen. Esto le provee del rendimiento más alto posible. Si hay cualquier error hizo un informe durante la compilación, los mensajes de error serán mostrados en el fondo del Redactor de NinjaScript.

-
o
-

Seminario: RSI con Objetivo de Ganancia & pérdida de Parada (el Nivel 2)

Descripción

En este seminario de nivel 2 vamos a construir la estrategia automatizada de una costumbre que va mucho tiempo cuando RSI se cruza encima 20 y salidas en un objetivo de ganancia o pérdida de parada predefinido, cualquiera es golpeado primero. Este seminario es demuestra el uso del Redactor de NinjaScript.

-o-

Establecer

El primer paso en la creación de una estrategia de encargo debe usar al mago de estrategia de encargo. El mago proporciona dos opciones:

- Permita usted para crear una estrategia funcional sin cualquier programación
- Genera el código de NinjaScript requerido que servirá de la fundación para su costumbre estrategia para codificación adicional

Este seminario cubrirá a éste, conscructing una estrategia vía la codificación directa en el redactor de NinjaScript.

1. Dentro de NinjaTrader Control Center ventana seleccionan los Instrumentos> Nuevo NinjaScript> Estrategia... el menú 2. Presione el botón "Next>"

p261

www.ninjatrader.co

m

© NinjaTrader, LLC,
2005

3. Entre en la información como mostrado encima 4. Presione el botón "Next>"

La definición de Parámetros de Entrada Debajo de usted definirá sus parámetros de entrada de estrategias. Éstos son cualquier parámetro de entrada que pueda ser cambiado por el usuario corriendo o backtesting una estrategia. Si su estrategia no requiere que cualquier parámetro deje los campos "de Nombre" en blanco.

p262

www.ninjatrader.co

m

© NinjaTrader, LLC,
2005

5. Añada las entradas según la imagen encima 6. Presione el botón "Unlock Code" para lanzar al Redactor de NinjaScript.

-o-

Entrada en Lógica de Estrategia

Usando Inicializar () Método de Configurar la Estrategia [Inicializar \(\)](#) el método es llamado una vez antes de la marcha de una estrategia y puede ser usado para poner propiedades o métodos de llamada en la preparación para dirigir una estrategia.

Entre en el código contenido dentro de Inicializar () método a la imagen abajo en Inicializar () método en el Redactor de NinjaScript.

[CalculateOnBarClose](#) es puesto al verdadero y pone la estrategia de llamar OnBarUpdate () método abajo en el final de cada barra en vez de cada marca de entrada. [Añada \(\)](#) el método es llamado y el [RSI \(\)](#) el método de indicador es pasado en que trazará automáticamente este indicador en una carta cuando la estrategia corre.

La firma de método para el RSI () indicador es:

[RSI \(período internacional, intervalo liso\)](#)

Es válido para haber usado llamado Añadir () el método en la manera siguiente:

Añada (RSI (14, 3))

Sin embargo, en vez de la codificación difícil del período valoran a 14 y el valor liso a 3, nosotros substituted los valores para el usuario definieron entradas que definimos en el mago. La vocación de Añadir () método en la manera siguiente:

Añada (RSI (RSIPeriod, RSISmooth))

Permite que nosotros cambiemos el período y los parámetros lisos del indicador RSI introducido en la estrategia en el tiempo de ejecución. Esto nos da un nivel más alto de flexibilidad trabajando con nuestra estrategia.

[SetStopLoss \(\)](#) y [SetProfitTarget \(\)](#) son llamados con CalculationMode. Marcas. Esto significa que cuando una posición es abierta, la estrategia presentará inmediatamente una parada y apuntará el pedido con un precio que es calculado basado en los parámetros de ProfitTarget y StopLoss pasados en la compensación del precio de entrada de promedio de posiciones.

Usando OnBarUpdate () Método para la Lógica de Estrategia Principal OnBarUpdate () el método es llamado para cada marca de entrada o en el final de una barra (usuario definido) realizando cálculos de tiempo real. Por lo tanto, esto es el método principal pidió el cálculo de estrategia y usaremos este método de entrar en la escritura que comprueban condiciones de salida y entrada.

p263

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Entre en el código contenido dentro de OnBarUpdate () método a la imagen abajo en OnBarUpdate () método en el Redactor de NinjaScript:

Ya que nuestra lógica de salida de estrategia ha sido establecida ya en Inicializar () método encima, sólo tenemos que concentrarnos en expresar nuestra lógica de entrada. La lógica de entrada de estrategia es muy honrada y puede ser traducida a inglés:

si no hemos visto que el número de barras especificadas por el usuario definió la entrada "RSIPeriod" entonces no van más lejos

*si RSI se cruza **encima de** un valor de 20 dentro de la última barra, vaya mucho tiempo*

Para llevar a cabo esto usamos los métodos siguientes y propiedades:

CurrentBar - Un valor que representa la barra corriente tratada (piensan en una carta donde la izquierda la mayor parte de barra sería igual a un) [CrossAbove \(\)](#) - los Controles de una cruz encima de la condición y devuelven [RSI](#) verdadero o falso [\(\)](#) - Vueltas el valor del indicador RSI [EnterLong \(\)](#) - Entra en un orden al precio de mercado mucho tiempo

-o-

Compilación

El código de estrategia es completo ahora y tiene que ser compilado.

- Si usted completara este seminario vía el Mago de Estrategia, simplemente siga las instrucciones de mago al final a cual tiempo la estrategia compilará.
 - Si usted mí cifrara este seminario usted puede compilar esta estrategia desde dentro NinjaScript
- El menú de menú "Compile" de botón de ratón de derecho de redactor o simplemente presiona la llave F5.

Es importante entender que este proceso hace la estrategia lista para el uso de tiempo real y correrá natalmente dentro de NinjaTrader directamente. Esto no corre interpretado tan muchas otras aplicaciones hacen. Esto le provee del rendimiento más alto posible. Si hay cualquier error hizo un informe durante la compilación, los mensajes de error serán mostrados en el fondo del Redactor de NinjaScript.

-o-

Referencia de Lengua de NinjaScript

Muestras de referencia & Puntas

p264

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Usted puede encontrar numerosas Muestras de Referencia de NinjaScript y Puntas en el Foro de Apoyo de NinjaTrader.

Haga clic aquí para [Muestras de Referencia](http://www.ninjatrader-support.com/vb/showthread.php?t=3220)
<<http://www.ninjatrader-support.com/vb/showthread.php?t=3220>>

Haga clic aquí para [Puntas](#)

-o-

Conceptos de Programación básicos

Empezar con NinjaScript

NinjaScript es una extensión del lenguaje de programación de Microsoft C#. Hay una riqueza de recursos disponibles en línea para aprender C# lengua. Si usted ha limitado la experiencia de programación, no deje C# asustarle. NinjaScript proporciona una capa de conveniencia que le permite concentrarse en sus ideas comerciales y no programación. Pase algún tiempo y pase por algunos de nuestros propios seminarios para desarrollar indicadores de encargo y estrategias.

Lo siguiente es un juego excelente del en línea C# seminarios [Acero Suave C# Seminarios](#).

-o-

Sintaxis básica

NinjaScript es una extensión a la lengua de Microsoft C# potente. El guía de sintaxis siguiente es un subconjunto de las capacidades del C# lengua. Para seminarios y una referencia en línea al C# visita de lengua la página [de descripción](#).

Las declaraciones Una declaración son análogas a una oración completar en la lengua inglesa.

Ejemplo:

Me gusta comerciar.

Esto está claro y fácil a entender. Si usted sólo escribiera:

Me gusta.

Cualquiera leyendo esto no sabría lo que le gusta. Esto no es una oración completar. La misma lógica se aplica a un C# declaración, forma una instrucción completa que puede ser interpretada por el compilador.

```
sume = 10 + 10;
```

La susodicha declaración es completa ya que decimos que la suma variable es igual a 10 más 10. Note que la declaración se termina con un punto y coma";" y no un período. En oraciones inglesas se terminan en períodos, en C#, final de declaraciones con un punto y coma.

Los Componentes básicos **de Componentes básicos son** análogos a párrafos en la lengua inglesa en esto ellos agrupan declaraciones juntos. En el mundo de programación llamamos esto un "Bloque del código". Estos bloques

p265

www.ninjatrader.com
© NinjaTrader, LLC, 2005

son delimitados con tirantes rizados " {}" como el ejemplo siguiente:

```
si (x == 5) {  
    Letra ("NinjaTrader"); Letra ("NinjaScript");  
}
```

El susodicho bloque de código encierra dos declaraciones en tirantes rizados.

Usuario Comentarios Definidos Usted puede añadir sus propios comentarios a su código.

Use "el/" caracteres para comentarios de línea solos.

Ejemplo:

```
//El código siguiente encierra dos declaraciones con tirantes rizados si (x == 5)  
{  
    Letra ("NinjaTrader"); Letra ("NinjaScript");  
}
```

Usted puede encerrar varias líneas de comentario usando el "/*" caracteres para comenzar el bloque de comentario y luego usando el "*/" caracteres para terminar el bloque de comentario.

Ejemplo:

```
/* Estos son comentarios para ilustrar un bloque de comentario de línea multi-dentro  
de NinjaScript */
```

La Sensibilidad de caso C# es una lengua sensible al caso el que significa que "NinjaTrader" con una capital "N" no es el mismo como "ninjaTrader" con una minúscula "n".

Las variables y los Tipos de Valor Una variable es un poseedor de lugar que almacena la información en la memoria del ordenador. Una variable es única análogo a su dirección

postal. Use variables para almacenar y tener acceso a datos. Hay muchos tipos variables que usted puede usar en el C# lengua. Lo siguiente es algunos de los fundamentos:

ensarte Tiendas datos textuales

doble valores de punto flotante de Tiendas

valores de número entero de Tiendas de número entero

Tiendas de bool verdaderas o falsas

p266

www.ninjatrader.com
© NinjaTrader, LLC, 2005

objeto

Objetos de tiendas, como indicadores NinjaTrader (en estos casos, usted declararía el tipo variable como el tipo de objeto sí mismo)

Declarando Variables declarar una variable en C# usted debe declarar primero su tipo de datos y luego proporcionar un nombre único y opcionalmente adjudicar un valor.

```
//La declaración de una variable de cuerda ensarta myString = "NinjaTrader";  
//La declaración de una doble variable dobla myDouble = 2.5;  
//Declaración de un intervalo de variable de número entero myInteger = 1;  
//Declaración de una variable bool bool learningIsFun = verdadero;  
//La declaración de un objeto escribe a máquina la variable usando un indicador  
Simple Moving Average SMA mySMA = nuevo SMA (Cerca, 20);
```

En cada uno de los susodichos ejemplos usted notará que el iguala el carácter "=" es usado para adjudicar un valor a la variable declarada.

Los operadores C# proporcionan un grupo grande de operadores, que son símbolos que especifican que operaciones funcionar en una declaración. Lo siguiente es un subconjunto de operadores comunes.

Aritmética

+	adición
-	substracción *
	multiplicación/ división

```
//Ejemplo de usar intervalo de operadores aritmético myInteger = 0; myInteger  
= 5 + (3 * 4); Letra (myInteger.ToString ());
```


El susodicho ejemplo imprimiría un valor de 17 a la ventana de salida de NinjaTrader. El Sistema. La clase de matemáticas proporciona funciones de matemáticas adicionales.

Por ejemplo:

```
//Ejemplo de usar el método ABS del Sistema. La clase de matemáticas dobla  
myDouble = Matemáticas. Abs (5 - 6); Letra (myDouble. ToString ());
```

Imprimiría un valor de 1 a la ventana de salida de NinjaTrader. Ver una lista completa de [Sytem. Métodos de matemáticas](#).

p267

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Lógico

&& y también || o sea

```
//Ejemplo de usar intervalo de operadores lógico myInteger = 3; ensarte myString =  
"NinjaTrader"; si (myInteger == 3 && myString == "NinjaTrader") {  
    Letra ("verdadera");}
```

El susodicho ejemplo imprimirá verdadero para la ventana de salida de NinjaTrader si la variable myString es igual a 3 y también (&& operador usado) la variable myString es igual a NinjaTrader.

Relacional

== != <> <=> =

es igual a no iguala menos que mayor que menos
que o igual al mayor que o igual a

```
//Ejemplo de usar cuerda de operador relacional myString = "NinjaTrader"; si  
(myString != "comerciendo") {  
    La letra ("Variable el myString no iguala el comercio");}
```

```
//El segundo ejemplo de usar a un operador relacional junto con el operador lógico  
dobla myDouble = 1000.25; si (myDouble <1000 || myDouble > 1001) {  
    La letra ("Variable el myDouble está entre 1000 y 1001");}
```

Asignación

= + = - = * = / =

Condicional

?:

igual a $x + = y$ es equivalente a $x = x + y$ $x - = y$ es
equivalente a $x = x - y$ $x * = y$ es equivalente a $x = x * y$
 $x / = y$ es equivalente a $x = x / y$

p268

www.ninjatrader.com
© NinjaTrader, LLC,
2005

```
//Ejemplo de un intervalo de operador condicional myInteger = (10 > 12 ? 3 : 4); Letra  
(MyInteger. ToString ());
```

La susodicha declaración condicional dice adjudican el valor 3 a la variable myInt si 10 es mayor que 12 más adjudican el valor 4 a la variable myInt. El ejemplo imprimirá entonces el valor de 4 a la ventana de salida de NinjaTrader ya que 10 no es mayor que 12.

El Encadenamiento de cuerda Para añadir una cuerda a otra cuerda usa "el +" carácter.

```
//El ejemplo del encadenamiento de cuerda ensarta wordOne = "Ninja"; ensarte wordTwo  
= "Comerciante"; Letra (MyInteger. ToString ());
```

El susodicho ejemplo iba la copia impresa de NinjaTrader a la ventana de salida de NinjaTrader.

o-

Órdenes que se bifurcan

Flujo de ejecución de mandos que se bifurca de su escritura. Esto permite que usted se bifurque en secuencias lógicas separadas basadas en decisiones que usted hace.

El si Declaración

Un *si* la declaración permite usted para tomar ejecuta caminos diferentes de la lógica según una condición dada.

```
//Intervalo de condición de caso solo x = 0; si (x == 0) {
    Letra ("NinjaTrader");}
```

El susodicho ejemplo imprimirá NinjaTrader a la ventana de salida de NinjaTrader ya que el x realmente iguala 0.

```
//El uno o el otro/O intervalo de decisión x = 1; si (x == 0) {
    Letra ("NinjaTrader");} más {
    Letra ("NinjaScript");}
}
```

p269

www.ninjatrader.com
© NinjaTrader, LLC, 2005

El susodicho ejemplo imprimirá NinjaScript a la ventana de salida de NinjaTrader.

```
//Intervalo de decisión de caso múltiple x = 2; si (x == 0) {
    Letra ("NinjaTrader");} más si (x == 1) {
    Letra ("NinjaScript");} más {
    Letra ("¡Reglas de NinjaTrader!");}
```

¡El susodicho ejemplo imprimirá Reglas de NinjaTrader! a la ventana de salida de NinjaTrader.

La Declaración de interruptor

La declaración de interruptor ejecuta un juego de la lógica según el valor de un parámetro dado.

```
//Intervalo de ejemplo de interruptor x = 2; cambie (x) {
    caso 0:
        La letra ("x es igual al cero"); ruptura;
    caso 1:
        La letra ("x es igual a un"); ruptura;
    caso 2:
        La letra ("x es igual a dos"); ruptura;
}
```

El susodicho ejemplo va la copia impresa x ser igual a dos a la ventana de salida de NinjaTrader.

-o-

Órdenes de colocación

La colocación manda el flujo de ejecución de control de su escritura. Si usted quisiera imprimir la palabra NinjaTrader 100 veces usted podrían usar 100 líneas del código o usted podría hacer la misma utilización de una orden de colocación en sólo unas líneas del código.

Mientras Lazo

mientras (Expresión booleana) {
 //Haga algo aquí}

Ejemplo:

p270

www.ninjatrader.com
© NinjaTrader, LLC, 2005

```
//Imprima NinjaTrader 100 veces al intervalo de ventana de salida x = 0; mientras (x  
<100) {  
    Letra ("NinjaTrader"); x = x + 1;  
}
```

Realmente Formar un lazo

haga {
 //Haga algo aquí} mientras (Expresión booleana)

Ejemplo:

```
//Imprima NinjaTrader 100 veces al intervalo de ventana de salida x = 0; haga {  
    Letra ("NinjaTrader"); x = x + 1;  
} mientras (x <100)
```

Para Lazo

para (initializer; expresión booleana; iterator) {
 //Haga algo aquí}

Ejemplo:

```
//Imprima NinjaTrader 100 veces a la ventana de salida para (intervalo x = 0; x <100;  
x ++ ) {  
    Letra ("NinjaTrader");}
```

Lazo de Foreach

el foreach (escriben a máquina el identificador en la expresión booleana) {
 //Haga algo aquí}

Ejemplo:

Deja dicen que usted quiso contar el número de galletas de harina de avena en una lata de galletas.

p271

www.ninjatrader.com
© NinjaTrader, LLC, 2005

```
//Cuenta el número de galletas de harina de avena en el intervalo de lata de galletas
oatmealCookies = 0; foreach (galleta en cookieJar) {
    si (galleta. Escriba a máquina == Harina de avena) {
        oatmealCookies = oatmealCookies + 1;
    } Letra ("Hay" + numberOatmeal. ToString () + "galletas de harina de avena
en la lata de galletas.");
```

Rompa y Siga

Usted puede usar la orden de ruptura de salir de un lazo en cualquier momento. El ejemplo siguiente es un lazo infinito pero nos romperemos después de la primera declaración de letra.

```
//Salga del lazo infinito después de la primera iteración mientras (0 == 0) {
    Letra ("NinjaTrader"); ruptura;
}
```

La orden seguir brincará delante a la siguiente iteración del lazo. El ejemplo siguiente no imprimirá NinjaTrader porque la orden seguir devuelve el control a la cumbre del lazo cada vez.

```
//Nunca letras NinjaTrader para (intervalo x = 0; x <100; x ++) {
    siga; Letra ("NinjaTrader");
}
```

-0-

Funciones y Métodos Explicados

Las funciones y los métodos son usados para encapsular un juego de declaraciones. Les dan un nombre y opcionalmente un juego de parámetros de entrada. Pueden llamarlos de cualquier punto en su NinjaScript. Una vez escrito, sólo tenemos que estar referidos por lo que la función o el método hacen. En general, la diferencia entre una función y método es que una función devuelve un valor a la rutina de vocación. C# ha suprimido el término la función y sólo usa el término el método. Un método puede devolver un valor (una función) o no devolver nada.

La declaración de un Método que Un método debe ser declarado primero antes de que usted pueda usarlo en su escritura.

[Acceso de método] [Tipo de datos de Vuelta] [Nombre] (Parámetros opcionales)

Acceso de método - Juegos el acceso al método. Este tema puede ser un pequeño complejo tan deja sólo al uso el nivel de acceso "privado" que es todo que usted necesitará probablemente alguna vez. Devuelva el Tipo de datos - Juegos el tipo de valor que la vuelta de método. Si esto no devuelve nada, usted pondrá el tipo de vuelta "al vacío".

p272

www.ninjatrader.com
© NinjaTrader, LLC,
2005

Nombre - Un usuario definió el nombre para el método Parámetros Opcionales - Cualquier parámetro opcional que usted pueda querer pasar en el método

Método Sin Tipo de Vuelta

```
//Este método imprime los datos y tiempo PrintDateTime vacío privado () {  
    Letra (DateTime. Ahora. ToString ());}
```

El susodicho método no tiene ningún tipo de vuelta entonces usamos la palabra clave "vacío", proveemos un usuario definió el nombre de método PrintDateTime y ya que no hay ningunos parámetros completamos la declaración con " ()". El código de método sí mismo es encerrado entonces dentro de tirantes rizados.

Método Con un Tipo de Vuelta

```
//Este método realiza un cálculo y vuelve un doble valor privado doble Se multiplican  
(doble entrada) {  
    devuelva la entrada * 10;}
```

El susodicho método devuelve un valor del tipo doble entonces usamos la palabra clave "doble", proveemos el nombre de método definido de un usuario "Se multiplican" y declaramos que este método toma "la entrada" llamada de un parámetro que es del tipo de valor doble. Ya que este método devuelve un valor, usamos la palabra clave "de vuelta" para devolver el valor de (entrada * 10) atrás a la rutina de vocación.

La vocación de un Método dentro de un Método

```
//Este método realiza un cálculo y vuelve un doble valor privado doble Se multiplican  
(doble entrada) {  
    PrintDateTime (); devuelva la entrada * 10;  
}
```

Añadiendo nuestros ejemplos desde encima, añadimos una llamada a PrintDateTime () método dentro de nuestro Multiplicarás el método.

Cuando usar métodos de Utilización **de Métodos** es una gran manera organizar lógicamente bloques del código. Si usted encuentra que realiza los mismos cálculos en partes diferentes de su escritura también tiene sentido de encapsular el código reiterativo en un método.

Por ejemplo:

Deja dicen que usted tenía una escritura que tenía que calcular la variedad media de las tres barras pasadas en posiciones múltiples. Esto es donde la encapsulación de la lógica a esto cuenta la variedad en un método entra práctico.

Primero declaramos nuestro método usando un tipo de vuelta del doble y lo llamamos AverageRange. No hay ningún requisito para pasar en cualquier parámetro. Entonces escribimos una declaración calcular y devuelve la variedad media de las tres barras pasadas. Terminamos con el método abajo.

p273

www.ninjatrader.com
© NinjaTrader, LLC, 2005

```
//Calcula la variedad media de las tres barras pasadas doble AverageRange privado
() {
    vuelva ((Alto [1] - Bajo [1]) + (Alto [2] - Bajo [2]) + (Alto [3] - Bajo [3]))
/ 3;}
```

Podemos referirnos entonces al método de AverageRange en cualquier otro sitio en nuestra escritura como el ejemplo debajo de espectáculos.

```
//El método de ejemplo que llama AverageRange () método ExampleMethod vacío dos veces
privado () {
    si (Alto [0] - Bajo [0] > AverageRange ()) {
        La letra ("La variedad de barra corriente es mayor que la 3 variedad de
promedio de barra
de "+
        AverageRange ().ToString ());
    }}
```

Declarando Variables dentro de un Método Usted puede declarar variables dentro de un método. Estas variables son locales en el alcance el que significa que sólo pueden tener acceso a ellos dentro del método y no fuera de él, como su escritura principal.

```
//Método de ejemplo usando doble MyMethod privado variable () {
    doble myDouble = 100.25 vuelva myDouble * 10;
}
```

Métodos de Estrategia de encargo y Propiedades

Acontecimiento avanzado Conducido Programando

Acontecimiento avanzado Conducido en

Programación de Estrategia

Descripción

La programación de estrategia conducida por el acontecimiento avanzada es reservada para programadores **CON EXPERIENCIA**.

NinjaTrader es a diferencia de plataformas de desarrollo de estrategia tradicionales en las cuales no es limitado con el procesamiento de su lógica de estrategia en una barra por la base de barra. NinjaTrader es una plataforma comercial de tiempo real verdadera y así puede ser conducido por acontecimientos de tiempo real.

Además de [OnBarUpdate \(\)](#) e Inicializan [\(\)](#) métodos, NinjaScript expone el siguiente

p274

www.ninjatrader.com

© NinjaTrader, LLC,

2005

métodos conducidos por el acontecimiento avanzados:

- [OnConnectionStatus \(\)](#) - Llamado cuando un estado de conexión cambia
- [OnExecution \(\)](#) - Llamado cuando una estrategia generó el pedido está lleno
- [OnMarketData \(\)](#) - Visitado cualquier cambio de una corriente de datos de mercado de nivel 1
- [OnMarketDepth \(\)](#) - Visitado cualquier cambio de una corriente de datos de mercado de nivel 2
- [OnOrderUpdate \(\)](#) - Llamado cuando una estrategia generó el estado de cambios de pedido
- [OnPositionUpdate \(\)](#) - Llamado cuando una estrategia generó cambios de posición

IExecution

La definición Representa un leído sólo conectan lo que expone la información en cuanto a una ejecución (llenar) resultar de un pedido y es pasado como un parámetro en [OnExecution](#)

() método.

* La propiedad "de Precios" puede devolver un valor que no es alineado a la talla de marca de instrumentos subyacente. Esto puede pasar si el precio de pedido original fuera presentado no alineado para señalar la talla o si el período en el tiempo siendo backtested contiene la hendidura/dividendo atrás ajustó datos que también no son alineados para señalar la talla. Sin embargo, el más si no todas las demostraciones de rejilla de ejecución mostrarán un doblado para señalar el valor de talla. Usted también puede usar el método [Round2Ticksize \(\)](#).

Métodos y Propiedades

Encargue Instrumento de ExecutionId MarketPosition

Señal de Tiempo de Cantidad de Precios de Pedido de nombre ToString ()

Ejemplos

Un [doble](#) valor que representa la comisión de una ejecución Un valor [de cuerda](#) que representa el cambio generó la ejecución id Un valor [de Instrumento](#) que representa el instrumento de un pedido valores Posibles es:

MarketPosition. Long MarketPosition. Corto

Un valor [de IOrder](#) que representa un pedido asociado a la ejecución Un [doble](#) valor que representa el precio de una ejecución Una cantidad de representación de valor [internacional](#) de una ejecución Una estructura [de DateTime](#) que representa el tiempo la ejecución ocurrió Una [cuerda](#) que representa id único de una ejecución Una representación [de cuerda](#) de una ejecución

```
IOrder privado entryOrder = nulo;

protegido anulan OnBarUpdate vacío () {
    si (entryOrder == nulo && Cerca [0]> Abierto [0])
        entryOrder = EnterLong ();}

protegido anulan OnExecution vacío (ejecución de IExecution) {
    si (!entryOrder && entryOrder.Señal == ejecución.Pedido.
    Señal)
        Letra (ejecución.ToString ());}
```

-o-

p275

www.ninjatrader.com
© NinjaTrader, LLC, 2005

IOrder

Definición

Representa un leído sólo conectan lo que expone la información en cuanto a un pedido.

- Un objeto de IOrder volvió de llamar un método de pedido es dinámico en aquel las propiedades de los it siempre reflejará que el estado actual de un pedido
- Un objeto de IOrder pasó en de OnOrderUpdate () el método es volátil en esto los its' las propiedades permanecen estáticas y no cambian. No agarre a estas referencias para la comprobación de las propiedades de los it fuera de OnOrderUpdate () el método. Compruebe que el objeto de IOrder volvió de la llamada original al método de pedido.
- Los objetos de IOrder no son **únicos** por pedido y por lo tanto no comparan objetos de IOrder con el uno al otro para determinar la igualdad
- Para comprobar la igualdad comprueban <IOrder>.Token la propiedad que es un único garantizado el identificador para un pedido
- La propiedad <IOrder>.OrderId no es **un** valor único ya que esto puede cambiar en todas partes un la vida del pedido

Métodos y Propiedades

Acción Los valores posibles son:

Acción. Compre Acción. Acción de BuyToCover. Venda Acción. SellShort
 AvgFillPrice Un **doble** valor que representa el promedio llena el precio de un pedido Lleno
 Un valor **internacional** que representa la cantidad llena de un pedido FromEntrySignal
 Una **cuerda** que representa al usuario definió el parámetro fromEntrySignal en un
 pedido Instrumento Un valor **de Instrumento** que representa el
 instrumento de un pedido LimitPrice Un **doble** valor que representa el precio de límite de un
 pedido
 LiveUntilCancelled *A **bool** representación de valor si el pedido es vivo hasta no anulado
 por el usuario
 Nombre
 Oco OrderId
 OrderState OrderType

Cantidad Tiempo de StopPrice TimeInForce

ToString simbólico ()

Una **cuerda** que representa el título de un pedido que puede ser proporcionado por la entrada o señal de salida llama Una **cuerda** que representa el OCO (uno anula otro) id de un pedido
 Una **cuerda** que representa al agente de bolsa publicó el pedido id valor (este valor puede cambiar) Ven que la mesa debajo de valores Posibles es:

OrderType. Límite OrderType. Mercado OrderType. Pare OrderType. StopLimit Un valor **internacional** que representa la cantidad de un pedido Un **doble** valor que representa el precio de parada de un pedido Una estructura de DateTime que representa la vez pasada que el pedido cambió valores Posibles estatales son:

TimeInForce. Día TimeInForce. Gtc Una representación [de cuerda](#)
<<http://msdn2.microsoft.com/en-us/library/system.datetime.aspx>> [id único de un pedido](#)
Una representación [de cuerda](#) de un pedido

p276

www.ninjatrader.com

© NinjaTrader, LLC,
2005

OrderState. OrderState aceptado. OrderState anulado. OrderState lleno. PartFilled

El Pedido **de Valores de OrderState** ha sido reconocido
por el Pedido de agente de bolsa ha sido anulado el Pedido ha estado
lleno el Pedido ha sido la parte llena
OrderState. PendingCancel Una solicitud de cancelación de pedido ha sido
presentado IOrderState. PendingChange Una solicitud de cambio de pedido ha
sido presentado OrderState ge. PendingSubmit Un pedido ha sido presentado
esto
OrderState. RechazadoUn pedido ha sido OrderState rechazado.
Funcionamiento Un pedido trabaja en OrderState de
cambio. Desconocido Un estado de pedido desconocido

*** En backtest histórico, los pedidos siempre alcanzarán un estado "Trabajador". ** En el de tiempo real, algunos pedidos de parada sólo pueden alcanzar el estado "Aceptado" si ellos son simulados/sostenidos en un servidor de agentes de bolsa**

Ejemplos

```
IOrder privado entryOrder = nulo;  
  
protegido anulan OnBarUpdate vacío () {  
    si (entryOrder == nulo && Cerca [0]> Abierto [0])  
        entryOrder = EnterLong ();}  
  
protegido anulan OnOrderUpdate vacío (pedido de IOrder) {  
    si (!entryOrder || entryOrder.Señal == pedido.Señal) {  
        Letra (pedido.ToString ()); si (pedido.OrderState == OrderState.Lleno)  
            entryOrder = nulo;  
    }  
}
```

Puntas 1. No todos los abastecedores de conectividad apoyan todo MarketDataTypes.

IPosition

La definición Representa un leído sólo conectan lo que expone la información en cuanto a una posición y es pasado como un parámetro en [OnPositionUpdate \(\)](#) método.

Métodos y Propiedades

AvgPrice Un **doble** valor que representa el precio medio de un Instrumento de posición
Un valor **de Instrumento** que representa el instrumento de una posición

p277

www.ninjatrader.com

MarketPosition

Cantidad ToString ()

Ejemplos

© NinjaTrader, LLC, 2005

Los valores posibles son:

MarketPosition. MarketPosition llano. Long MarketPosition. Corto Una cantidad de representación de valor **internacional** de una posición Una representación **de cuerda** de una posición

```
protegido anulan OnPositionUpdate vacío (posición de IPosition) {  
    si (posición. MarketPosition == MarketPosition. Piso) {  
        //Haga le gusta reinicializa algunas variables aquí  
    }  
}
```

OnConnectionStatus ()

La definición OnConnectionStatus () método es llamada en un cambio del estado de conexión.

- La programación a este método se considera la programación **avanzada** y expuesta para los programadores con experiencia
- Usted no pueden comenzar una estrategia sin una conexión "Relacionada"
- El acontecimiento de conexión es para la conexión donde la cuenta cambiada reside

El Valor de retorno de método Este método no devuelve un valor.

Parámetros de método `ConnectionStatus` `orderStatus` y `priceStatus`

Los valores posibles son:

`ConnectionStatus`. `ConnectionStatus` relacionado. Unión de `ConnectionStatus`. `ConnectionLost`
`ConnectionStatus`. Deshilvanado

Una conexión de NinjaTrader puede tener dos internamente puede tener dos conexiones separadas, un para el servidor de precios y otro para el servidor de pedido.

`orderStatus` - Estado del servidor de pedido del agente de bolsa `priceStatus` - Estado del servidor de precios

Sintaxis Usted debe anular el método en su estrategia con la sintaxis siguiente.

```
protegido anulan OnConnectionStatus vacío (ConnectionStatus orderStatus,  
ConnectionStatus priceStatus) {  
  
}
```

p278

www.ninjatrader.com
© NinjaTrader, LLC,

2005

Ejemplos

```
ConnectionStatus privado dataFeed = ConnectionStatus. Relacionado;  
  
protegido anulan OnOrderUpdate vacío (pedido de IOrder) {  
    si (!dataFeed! = ConnectionStatus. Relacionado) {  
        //Haga algo  
    }  
}  
  
protegido anulan OnConnectionStatus vacío (ConnectionStatus orderStatus,  
ConnectionStatus priceStatus) {  
    dataFeed = priceStatus;}  
}
```

Las Muestras de Referencia adicionales muestras de código de referencia Adicionales están disponibles NinjaScript sección de Recursos Educativa de nuestro foro de apoyo.

-0-

OnExecution ()

La definición `OnExecution ()` método es llamada en una ejecución de entrada. Una ejecución es otro nombre para llenar de un pedido.

- La programación a este método se considera la programación **avanzada** y expuesta

para

los programadores con experiencia • Un pedido pueden generar ejecuciones múltiples (parcial se llena) • OnExecution siempre es llamado después de que [OnOrderUpdate \(\)](#) es llamado

Si usted confía en OnExecution () el método de provocar acciones, como la sumisión de un pedido de pérdida de parada cuando su pedido de entrada **SIEMPRE está** lleno se refieren a las propiedades en la propiedad de objeto [de IOrder](#) atada al objeto de IExecution pasado en OnExecution () método.

El Valor de retorno de método Este método no devuelve un valor.

Parámetros de método ejecución [de IExecution](#)

Sintaxis Usted debe anular el método en su estrategia con la sintaxis siguiente.

```
protegido anulan OnExecution vacío (ejecución de IExecution) {  
  
}
```

Ejemplos

p279

www.ninjatrader.co

m

© NinjaTrader, LLC,
2005

```
IOrder privado entryOrder = nulo;  
  
protegido anulan OnBarUpdate vacío () {  
    si (entryOrder == nulo && Cerca [0]> Abierto [0])  
        entryOrder = EnterLong ();}  
  
protegido anulan OnExecution vacío (ejecución de IExecution) {  
    si (!entryOrder && entryOrder. Señal == ejecución. Pedido. Señal)  
        Letra (ejecución. ToString ());}
```

Las Muestras de Referencia adicionales muestras de código de referencia Adicionales están disponibles NinjaScript sección de Recursos Educativa de nuestro foro de apoyo.

-o-

OnMarketData ()

Ver [OnMarketData \(\)](#)

-o-

OnMarketDepth ()

Ver [OnMarketDepth \(\)](#)

-o-

OnOrderUpdate ()

Definición OnOrderUpdate () el método es llamado cada vez un pedido manejado por una estrategia cambia el estado. Un pedido cambiará el estado cuando un cambio de cantidad de pedido, precio o estado (trabajando al lleno) ocurre.

- La programación a este método se considera la programación **avanzada** y expuesta para los programadores con experiencia • Usted pueden programar su propio [manejo de rechazo de pedido](#)

NinjaTrader es una aplicación multitenhebrada y por lo tanto es muy importante entender

- Este método garantiza que usted verá que cada cambio de estado de pedido de la secuencia • Este método no proporciona una actualización al estado más actual de un pedido, pero en cambio le provee un acontecimiento que le notifica de cada cambio estatal de la secuencia y la información relevante a la orden entonces el estado cambió
- **CRÍTICO:** Si usted quiere ir en coche su lógica de estrategia basada en el pedido se llena usted debe usar [OnExecution \(\)](#) en vez de OnOrderUpdate (). OnExecution () siempre es provocado después de OnOrderUpdate (). Hay lógica de estrategia interna que es provocada después de que OnOrderUpdate () es llamado, pero antes de OnExecution () que puede afectar negativamente su estrategia si usted confía en el rastreo se llena dentro de OnOrderUpdate ().
- Como un ejemplo, el corazón de NinjaTrader puede haber recibido "el Funcionamiento" y luego "PartFilled" pida acontecimientos de cambio estatales atrás del agente de bolsa API en el hilo "B" y a algún punto a tiempo (milisegundos más tarde) el corazón de NinjaTrader tomará estos acontecimientos y provocará OnOrderUpdate () método en la estrategia del hilo un. Así, cuando la estrategia recibe

p280

www.ninjatrader.com
© NinjaTrader, LLC, 2005

- el primer estado "Trabajador" para un pedido, el objeto de IOrder pasó en reflejará el estado "Trabajador" aunque el pedido actual esté realmente en un estado de "la Parte Llena" que es realmente reflejado en el objeto de IOrder original devuelto en cualquier de los métodos de pedido, como [EnterLong \(\)](#). Por supuesto, OnOrderUpdate () método recibirá posteriormente el acontecimiento para el estado "de PartFilled".
- Cuando relacionado con la [Conexión de Repetición de Mercado](#), llamando orden al precio de mercado métodos basados tal ya que EnterLong () y EnterShort () causará acontecimientos de estado de pedido disparados antes de la vuelta de método de pedido un objeto de IOrder. Esto es hecho para asegurar que todos los acontecimientos están en la sincronización en repeticiones de

mercado de alta velocidad.

Si usted confía en `OnOrderUpdate ()` el método de provocar acciones, como la sumisión de un pedido de pérdida de parada cuando su pedido de entrada **SIEMPRE está** lleno se refieren a las propiedades en el objeto de `IOrder` pasado en `OnOrderUpdate ()` método.

El Valor de retorno de método Este método no devuelve un valor.

Parámetros de método pedido [de IOrder](#)

Sintaxis Usted debe anular el método en su estrategia con la sintaxis siguiente.

```
protegido anulan OnOrderUpdate vacío (pedido de IOrder) {  
  
}
```

Ejemplos

```
IOrder privado entryOrder = nulo;  
  
protegido anulan OnBarUpdate vacío () {  
    si (entryOrder == nulo && Cerca [0]> Abierto [0])  
        entryOrder = EnterLong ();  
  
protegido anulan OnOrderUpdate vacío (pedido de IOrder) {  
    si (!entryOrder && entryOrder.Señal == pedido.Señal) {  
        Letra (pedido.ToString ()); si (pedido.OrderState == OrderState.Anulado)  
        {  
            //Haga algo aquí entryOrder = nulo;  
        }  
    }  
}}
```

Las Muestras de Referencia adicionales muestras de código de referencia Adicionales están disponibles NinjaScript sección de Recursos Educativa de nuestro foro de apoyo.

-o-

OnPositionUpdate ()

Definición

p281

www.ninjatrader.com
© NinjaTrader, LLC, 2005

`OnPositionUpdate ()` el método es llamado cada vez una posición manejada por una estrategia cambia el estado.

- La programación a este método se considera la programación **avanzada** y expuesta para programadores **con experiencia**

El Valor de retorno de método Este método no devuelve un valor.

Parámetros de método posición de [IPosition](#)

Sintaxis Usted debe anular el método en su estrategia con la sintaxis siguiente.

```
protegido anulan OnPositionUpdate vacío (posición de IPosition) {  
}
```

Ejemplos

```
protegido anulan OnPositionUpdate vacío (posición de IPosition) {  
    si (posición. MarketPosition == MarketPosition. Piso) {  
        //Haga le gusta reinicializa algunas variables aquí  
    }  
}
```

Las Muestras de Referencia adicionales muestras de código de referencia Adicionales están disponibles NinjaScript sección de Recursos Educativa de nuestro foro de apoyo.

-O-

Métodos de AtmStrategy

AtmStrategyCancelEntryOrder ()

La definición Anula el pedido de entrada especificado.

Las Vueltas **de Valor de retorno de método** verdaderas si el pedido especificado fuera encontrado; por otra parte **falso**.

[AtmStrategyCancelEntryOrder](#) **de sintaxis** (ensartan *orderId*)

Parámetros

limitPrice	Precio de límite de pedido	stopPrice	Precio
de parada de pedido	orderId	El identificador único para el	
pedido de entrada			

Ejemplos

p282

www.ninjatrader.com
© NinjaTrader, LLC, 2005

```
protegido anulan OnBarUpdate vacío () {  
    AtmStrategyCancelEntryOrder ("orderIdValue");  
}
```

-O-

AtmStrategyChangeEntryOrder ()

Cambios **de definición** el precio del pedido de entrada especificado.

Las Vueltas **de Valor de retorno de método** verdaderas si el pedido especificado fuera encontrado; por otra parte **falso**.

AtmStrategyChangeEntryOrder de sintaxis (doblan *limitPrice*, doblan *stopPrice*, ensartan *orderId*)

Parámetros

limitPrice	Precio de límite de pedido	stopPrice	Precio de
parada de pedido	orderId	El identificador único para el	
pedido de entrada			

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    AtmStrategyChangeEntryOrder (GetCurrentBid (), 0, "orderIdValue");  
}
```

-O-

AtmStrategyChangeStopTarget ()

Cambios **de definición** el precio del pedido especificado de la estrategia de ATM especificada.

Las Vueltas **de Valor de retorno de método** verdaderas si el pedido especificado fuera encontrado; por otra parte **falso**.

AtmStrategyChangeStopTarget de sintaxis (doblan *limitPrice*, doblan *stopPrice*, ensartan *orderName*, ensartan *AtmStrategyId*)

Parámetros

limitPrice	stopPrice	orderName	AtmStrategyId
------------	-----------	-----------	---------------

Ejemplos

La parada de Pedido de precios de límite de pedido valora El nombre de pedido, como "el STOP1" o "TARGET2" El identificador único para la estrategia de ATM

```
protegido anulan OnBarUpdate vacío () {  
    AtmStrategyChangeStopTarget (0, SMA (10) [0], "STOP1", "AtmIdValue");  
}
```

-O-

AtmStrategyClose ()

La definición Anula cualquier orden de trabajo y cierra cualquier puesto vacante de una estrategia por la primera modificación de cualquier pedido objetivo existente entonces ellos están llenos o sometimiento de un orden al precio de mercado si ningunos pedidos objetivo existen.

Las Vueltas **de Valor de retorno de método** verdaderas si la estrategia de ATM especificada fuera encontrada; por otra parte **falso**. Verdadero de NINGUNA MANERA indica que la estrategia de hecho ha estado cerrada. Esto indica que ATMstrategy especificado fue encontrado y la rutina cercana interna fue provocada.

AtmStrategyClose de sintaxis (ensartan *AtmStrategyId*)

Parámetros

AtmStrategyId El identificador único para la estrategia de ATM

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    de Ser (histórico)  
        vuelva;  
  
    //Compruebe la condición válida y cree una Estrategia de ATM si  
    (GetAtmStrategyUnrealizedProfitLoss ("idValue") > 500)  
        AtmStrategyClose ("idValue");  
}
```

-O-

AtmStrategyCreate ()

La definición Presenta un pedido de entrada que ejecutará la Estrategia de ATM especificada que puede ser manejada entonces a mano por cualquier ventana de entrada de pedido, como SuperDOM o dentro de su estrategia NinjaScript.

- Por favor examine la sección en [la utilización de Estrategias de ATM](#) • Este método no está de vuelta verificable y no ejecutará en datos históricos • Las ejecuciones que resultan de la Estrategia de ATM no serán trazadas en la carta • Ver [AtmStrategyCancelEntryOrder \(\)](#) para anular un pedido de entrada • Ver [AtmStrategyChangeEntryOrder \(\)](#) para cambiar el precio del pedido de entrada

Las Vueltas **de Valor de retorno de método** verdaderas si la estrategia de ATM especificada fuera creada y ningunos errores fueron generadas.

AtmStrategyCreate de sintaxis (*Acción de acción*, *OrderType orderType*, *doble limitPrice*, *doblan stopPrice*, *TimeInForce timeInForce*, *ensartan orderId*, *ensartan strategyTemplateName*, *ensartan AtmStrategyId*)

p284

www.ninjatrader.com
© NinjaTrader, LLC, 2005

acción

orderType

limitPrice stopPrice timeInForce

orderId

Los Juegos **de parámetros** si el pedido de entrada es una compra u orden de venta. Los valores posibles son:
Acción. Compre la Acción. Venda Juegos el tipo de pedido del pedido de entrada. Los valores posibles son: OrderType. Límite OrderType. Mercado OrderType. Pare OrderType.
StopLimit El precio de límite del pedido El precio de parada del pedido Juegos el tiempo vigente del pedido de entrada. Los valores posibles son: TimeInForce. Día TimeInForce.
Gtc El identificador único para el pedido de entrada
el strategyTemplateN Especifica qué plantilla de estrategia será usada
ame
AtmStrategyId El identificador único para la estrategia de ATM

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    de Ser (histórico)  
        vuelva;  
  
    //Compruebe la condición válida y cree una Estrategia de ATM si (Cerca [0]> SMA  
    (20) [0])  
        AtmStrategyCreate (Acción. Compre, OrderType. Mercado, 0, 0,  
            TimeInForce. Día, GetAtmStrategyUniqueId (), "MyTemplate",  
            GetAtmStrategyUniqueId ());  
}
```

-o-

GetAtmStrategyEntryOrderStatus ()

La definición Consigue el estado actual del pedido de entrada especificado. Si el método no puede encontrar el pedido especificado, una serie vacía es devuelta.

Valor de retorno de método Una [cuerda \[\]](#) la serie sosteniendo tres elementos que representan el promedio llena el precio, la cantidad llena y [pide el estado](#).

[GetAtmStrategyEntryOrderStatus](#) **de sintaxis** (ensartan *orderId*)

Parámetros

orderId El identificador único para el pedido de entrada

Ejemplos

p285

www.ninjatrader.com
© NinjaTrader, LLC, 2005

```
protegido anulan OnBarUpdate vacío () {  
    cuerda [] entryOrder = GetAtmStrategyEntryOrderStatus ("orderId");  
  
    //Compruebe la longitud para asegurar que la serie devuelta sostiene la  
    información de pedido si (entryOrder.Longitud > 0) {  
        La letra ("Promedio llenan el precio es" + entryOrder [0].ToString ());  
        la Letra ("Cantidad llena es" + entryOrder [1].ToString ()); la Letra  
        ("estado actual es" + entryOrder [2].ToString ());  
    }  
}
```

-o-

GetAtmStrategyMarketPosition ()

La definición Consigue la posición en el mercado corriente de la Estrategia de ATM especificada.

Valor de retorno de método MarketPosition. MarketPosition llano. Long MarketPosition. Corto

[GetAtmStrategyMarketPosition](#) **de sintaxis** (ensartan *AtmStrategyId*)

Parámetros

AtmStrategyId El identificador único para la estrategia de ATM

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    //Compruebe si piso si (GetAtmStrategyMarketPosition ("id") ==  
    MarketPosition. Piso)  
        La letra ("posición de Estrategia de ATM es actualmente llano");}
```

-o-

GetAtmStrategyPositionAveragePrice ()

La definición Consigue el precio medio de la situación actual de la Estrategia de ATM especificada.

Valor de retorno de método Un **doble** valor que representa el precio medio.

GetAtmStrategyPositionAveragePrice de sintaxis (ensartan *AtmStrategyId*)

Parámetros

AtmStrategyId El identificador único para la estrategia de ATM

p286

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    //Compruebe si piso si (;GetAtmStrategyMarketPosition ("id")! =  
    MarketPosition. Piso)  
        La letra ("Precio medio es" + GetAtmStrategyPositionAveragePrice  
    ("id").ToString ());}
```

-o-

GetAtmStrategyPositionQuantity ()

La definición Consigue la cantidad de situación actual de la Estrategia de ATM especificada.

Valor de retorno de método Un valor **internacional** que representa la cantidad.

GetAtmStrategyPositionQuantity de sintaxis (ensartan *AtmStrategyId*)

Parámetros

AtmStrategyId El identificador único para la estrategia de ATM

Ejemplos

```

protegido anulan OnBarUpdate vacío () {
    //Compruebe si piso si (!GetAtmStrategyMarketPosition ("idValue") !=
    MarketPosition. Piso)
        La letra ("Talla de posición es" + GetAtmStrategyPositionQuantity
        ("id").ToString ());}

```

-0-

GetAtmStrategyRealizedProfitLoss ()

La definición Consigue el valor de ganancias y pérdidas realizado de la Estrategia de ATM especificada.

Valor de retorno de método Un **doble** valor que representa las ganancias y pérdidas realizadas.

GetAtmStrategyRealizedProfitLoss de sintaxis (ensartan *AtmStrategyId*)

Parámetros

AtmStrategyId El identificador único para la estrategia de ATM

Ejemplos

```

protegido anulan OnBarUpdate vacío () {
    La letra ("PnL es" + GetAtmStrategyRealizedProfitLoss ("id").ToString ());}

```

p287

www.ninjatrader.com
© NinjaTrader, LLC, 2005

-0-

GetAtmStrategyStopTargetOrderStatus ()

La definición Consigue el estado (s) actuales de la parada especificada o pedido objetivo. Si el método no puede encontrar el pedido (s) especificado, una serie vacía es devuelta.

!!!! Una parada especificada o el objetivo dentro de una estrategia de ATM realmente pueden sostener pedidos múltiples. Por ejemplo, si su estrategia de ATM presenta una parada y objetivo y usted recibe múltiple parcial se llena en la entrada de una tardanza de unos segundos o más entre la entrada se llena, la estrategia de ATM presentará la parada y apuntará pedidos por cada uno parcial llenan todos del mismo precio y piden el tipo.

Valor de retorno de método Una **cuerda []** serie multidimensional sosteniendo tres dimensiones que representan el promedio llena el precio, la cantidad llena y **pide el estado**. La longitud (número de elementos) representa el número de pedidos que representan el nombre especificado.

GetAtmStrategyStopTargetOrderStatus de sintaxis (ensartan *orderName*, ensartan *AtmStrategyId*)

Parámetros

orderName El nombre de pedido, como "STOP1" o
AtmStrategyId "TARGET2" El identificador único para la
estrategia de ATM

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    cuerda [] pedidos = GetAtmStrategyStopTargetOrderStatus ("TARGET1",  
"idValue");  
  
    //Compruebe la longitud para asegurar que la serie devuelta sostiene la  
información de pedido si (pide. Longitud > 0) {  
        para (intervalo i = 0; yo <pedidos. GetLength (0) - 1; yo++) {  
            La letra ("Promedio llenan el precio es" + pedidos [yo,  
0].ToString ()); la Letra ("Cantidad llena es" + pedidos [yo,  
1].ToString ()); la Letra ("estado actual es" + pedidos [yo,  
2].ToString ());  
        }  
    }  
}
```

-0-

GetAtmStrategyUnrealizedProfitLoss ()

La definición Consigue el valor de ganancias y pérdidas no realizado de la Estrategia de ATM especificada.

Valor de retorno de método Un **doble** valor que representa las ganancias y pérdidas no realizadas.

GetAtmStrategyUnrealizedProfitLoss **de sintaxis** (ensartan *AtmStrategyId*)

Parámetros

p288

www.ninjatrader.com
© NinjaTrader, LLC, 2005

AtmStrategyId El identificador único para la estrategia de ATM

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    La letra ("PnL no realizado es" +  
GetAtmStrategyUnrealizedProfitLoss ("id").ToString  
());}
```

-0-

GetAtmStrategyUniqueId ()

La definición Genera un valor de id único.

Valor de retorno de método Un valor [de cuerda](#) que representa un valor de id único.

Sintaxis GetAtmStrategyUniqueId ()

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    ensarte orderId = GetAtmStrategyUniqueId ();  
}
```

-O-

Métodos de pedido

Descripción

Los métodos de pedido son envueltos con una capa de conveniencia invisible que permite que usted se concentre en las reglas comerciales de su sistema dejando la mecánica subyacente de la dirección de pedido y las relaciones entre entrada y pedidos de salida y posiciones a NinjaTrader. Esto se hará más evidente ya que usted examina esta sección.

Unos puntos claves:

- Los pedidos son presentados como órdenes de trabajo vivas cuando una estrategia dirige en • de tiempo real *Profit objetivos, pare la pérdida y los pedidos de parada de rastro son presentados inmediatamente como pedidos de entrada
- están llenos y también son los cambios de Pedido de • atados de OCO y las cancelaciones son hechas cola en caso de que el pedido esté en un estado donde no puede ser anulado o modificado • en Ausencia, pedidos presentados vía la Entrada () y Salida () los métodos automáticamente anulan en el
- el final de una barra si no • presentado de nuevo los programadores Con experiencia pueden presentar pedidos que son vivos hasta no anulado

* Vía [SetProfitTarget \(\)](#), [SetStopLoss \(\)](#) y [SetTrailStop \(\)](#) métodos

Sumisión de pedido para Métodos de Salida y Entrada - Operación Básica

p289

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Usted puede presentar y manejar pedidos de uno de dos modos:

- Los pedidos que automáticamente expiran (anulan) al final de barra a menos que no

presentado de nuevo • Pedidos que son vivos hasta que usted deliberadamente los anule

El primer y la mayor parte de acercamiento común son el enfoque más rápido y más fácil y son convenientes para programadores de todos los niveles. El segundo enfoque, lo que llamamos vivo hasta no anulado, es reservado para programadores con experiencia sólo.

Los Pedidos de Acercamiento común son presentados principalmente desde dentro [OnBarUpdate \(\)](#) método cuando se llama un método de pedido específico. En ausencia, los pedidos son guardados vivos a condición de que ellos sean presentados de nuevo en cada llamada de [OnBarUpdate \(\)](#) método. Si el pedido no es presentado de nuevo, el pedido es anulado entonces. Si el pedido es presentado de nuevo con parámetros cambiados (nuevo precio de límite por ejemplo) el pedido es modificado.

En el ejemplo abajo, un orden límite de compra para entrar en una posición larga trabaja en el precio de oferta a condición de que el precio cercano de la barra corriente sea mayor que el valor corriente del 20 período el valor medio móvil simple. Si la condición de entrada ya no es verdad y el pedido todavía es activo será inmediatamente anulado.

```
protegido anulan OnBarUpdate vacío () {  
    //Condición de entrada si (Cerca [0]> SMA (20) [0])  
    EnterLongLimit (Barras. CurrentBid);}
```

El Enfoque Avanzado Los abastecedores de enfoque avanzados un enfoque más granular para pedir la sumisión y la dirección que puede ser útil si usted decide distribuir su lógica de estrategia fuera de [OnBarUpdate](#) tradicional () se acerca con otro acontecimiento de estrategia conducido en métodos, como [OnOrderUpdate \(\)](#) y [OnMarketData \(\)](#) sólo para llamar a unos cuantos. Después de examinar esta sección de descripción en Métodos de Pedido por favor examine el [Pedido Avanzado que Maneja](#) la sección para más información sobre el enfoque avanzado para pedir la sumisión.

Los métodos de Entrada **de Métodos de entrada son** usados para presentar pedidos que crean una posición en el mercado si ninguno existe o invertir una posición existente. Un ejemplo de un método de entrada que presentó un orden al precio de mercado de compra parecería a lo siguiente:

```
EnterLong ();
```

Nombres de señal en Métodos de Entrada Usted puede etiquetar opcionalmente un pedido de entrada con un nombre de señal. Los nombres de señal son usados para identificar ejecuciones que resultan del pedido de una carta y en informes de rendimiento. Las posiciones en el mercado creadas de un método de entrada etiquetado son marcadas con el nombre de señal que sirve dos objetivos:

- Usado para atar un método de salida a una posición específica
- Usado para identificar entradas únicas en una estrategia

Un ejemplo de un método de entrada que presentó vender el orden al precio de mercado corto con un nombre de señal etiquetado parecería a lo siguiente:

EnterShort ("Mi Nombre de Señal");

a la etiqueta de Estrategias de la ventana Control Center.

- La propiedad [de EntriesPerDirection](#) - Establece el número máximo de entradas en una **Definición** de dirección sola **como los Métodos de Entrada son Tratados en una Estrategia que** Usted puede limitar cuantos métodos de entrada son tratados determinando el número máximo de entradas en una dirección sola a través de todos los métodos de entrada, o a través de métodos de entrada únicos, definidos proporcionando nombres de señal. Las propiedades siguientes pueden ser puestas en la ventana Strategy Dialog añadiendo una estrategia a una carta o propiedad [de EntryHandling](#) • - Límites propiedad de EntriesPerDirection a través de todas las entradas en por entradas únicas (etiquetado con un nombre de señal)

Para ilustrar como las susodichas propiedades controlan el procesamiento de métodos de entrada van a mirar el ejemplo de código abajo. El código contiene dos condiciones de entrada y dos métodos de EnterLong cada uno etiquetado con nombres de señal únicos.

```
protegido anulan OnBarUpdate vacío () {
    //Condición de entrada 1 si (CrossAbove (SMA (10), SMA (20), 1))
        EnterLong ("Condición 1 Entrada");

    //Condición de entrada 2 si (CrossAbove (RSI (14, 3), 30, 1))
        EnterLong ("Condición 2 Entrada");}
```

Si usted quisiera su estrategia **sólo** de entrar **una vez** por dirección en condición 1 o en condición 2, cualquier condición evaluada al verdadero primero, usted pondría EntriesPerDirection a un valor de "1" y EntryHandling a un valor "AllEntries".

Si usted quisiera su estrategia sólo de entrar una vez por dirección para ambas condiciones, usted pondría EntriesPerDirection a un valor de "1" y EntryHandling al valor "UniqueEntries". Esto es posible ya que los métodos de entrada han sido etiquetados con nombres de señal únicos.

Los Métodos de entrada en Estrategias de Multiinstrumento dirigiendo estrategias que presentan pedidos a instrumentos múltiples, métodos de entrada presentarán pedidos al instrumento referido por [BarsInProgress](#). El ejemplo siguiente supone que la estrategia dirige durante un 1 minuto la carta de ES. Esto añade una serie de datos ER2 y entra en una posición tanto en el ES como en ER2.

```
protegido anulan el vacío Inicializan () {
    Añada (GetInstrument ("ER2 12-06", PeriodType. Minuto, 1);}

protegido anulan OnBarUpdate vacío () {
    si (BarsInProgress == 0)
```

```

EnterLong ("Comercio de ES");
más si (BarsInProgress == 1)
EnterLong ("Comercio de ER2");
}

```

p291

www.ninjatrader.co

m

© NinjaTrader, LLC,

2005

Tipo de cantidad y TIF

Usted también puede poner la cantidad de pedido de entrada y pedir el tipo que es generalmente puesto vía el UI en el tiempo de ejecución.

- La propiedad de QuantityType - Juegos si la cantidad de pedido es tomada de la cantidad de método de entrada
- propiedad o la talla de cantidad de estrategia de falta
- [TimeInForce](#) property - Juegos el tiempo vigente del pedido

Cerrando una Posición usando una Pérdida de Parada, Rastreo de Parada y/o

Objetivo de Ganancia Usted puede predefinir una pérdida de parada, arrastrando la parada y/o sacar ganancia el objetivo en una estrategia llamando [SetStopLoss \(\)](#), [SetTrailStop \(\)](#) o [SetProfitTarget \(\)](#) métodos de dentro de [Inicializar \(\)](#) método. Cuando estos métodos son llamados, ellos presentan órdenes de trabajo vivas en la operación de tiempo real ya que las ejecuciones son relatadas a consecuencia de la vocación de un método de entrada. Párese y apunte pedidos son presentados al mercado tan pronto como se llena entra de un pedido de entrada. Los pedidos de parada rastreadores son modificados en el de tiempo real y **NO** en el final de una barra. Una parada y el pedido objetivo también son atados vía OCO (uno anula el otro) tan si uno está lleno, el otro es automáticamente anulado.

Las paradas y los objetivos son generados por se llenan o por posición. Esto es determinado por la "Parada & propiedad" de sumisión objetivo que es puesta dirigiendo una estrategia viva vía el UI. Los valores de propiedad son:

ByStrategyPosition - Cuando esto es seleccionado, sólo una pérdida de parada, parada de rastreo y/o pedido objetivo de ganancia es presentado. Como las ejecuciones de entrada entran, la talla de pedido es enmendada. La desventaja de este enfoque es que si usted tiene parcial se llena, los pedidos son insertados de nuevo en la cola de pedido de cambio. El aspecto positivo es que si usted hace corretaje le acusa un por comisión de pedido (no por parte), usted no incurrirá en gastos de comisión adicionales.

PerEntryExecution - Cuando esto es seleccionado, una pérdida de parada, parada de rastreo y/o pedido objetivo de ganancia es presentado para cada uno parcial se llenan recibido. La desventaja es si su agente de bolsa acusa la comisión por pedido, usted puede incurrir en gastos de comisión muy caros si usted recibe parcial se llena. El aspecto positivo es que los pedidos son presentados cuanto antes dándole la ventaja de entrar en la cola de pedido inmediatamente.

El cierre de una Posición usando unos métodos de Salida **de Método de Salida** presenta pedidos de liquidar una posición en el entero o en parte. Un ejemplo de un método de salida que presentó un orden al precio de mercado vender para cerrar la posición de estrategia entera parecería a lo siguiente:

```
ExitLong ();
```

Cerrando una Posición Parcial usando un Método de Salida Usted puede liquidar una posición parcial especificando la cantidad de salida. El ejemplo siguiente va mucho tiempo 3 contratos. Cada actualización de barra subsecuente presentará un orden al precio de mercado para salir de un contrato hasta que la posición esté completamente cerrada. ExitLong (1) no será ignorado si una posición en el mercado larga no existe.

```
protegido anulan OnBarUpdate vacío () {  
    si (CrossAbove (SMA (10), SMA (20), 1))  
        EnterLong (3);  
  
    ExitLong (1);  
}
```

p292

www.ninjatrader.com
© NinjaTrader, LLC, 2005

La atadura de una Salida a una Entrada: Como Cerrarse una Posición Etiquetada con unas entradas de Identificación **de Nombre de Señal** con un nombre de señal permite que usted tenga entradas únicas múltiples dentro de una estrategia sola y métodos de salida de llamada con nombres de señal especificados de modo que sólo una posición creada con el nombre de señal etiquetado esté cerrada. En el ejemplo abajo, hay dos condiciones de entrada que crean posiciones. Hay dos condiciones de salida que especifican qué posición cerrarse basado en la señal llaman.

```
protegido anulan OnBarUpdate vacío () {  
    //Condición de entrada 1 si (CrossAbove (SMA (10), SMA (20), 1))  
        EnterLong ("Condición 1 Entrada");  
  
    //Condición de entrada 2 si (CrossAbove (RSI (14, 3), 30, 1))  
        EnterLong ("Condición 2 Entrada");  
  
    //Cierra la posición creada por la condición de entrada 1 si (CrossBelow (SMA  
    (10), SMA (20), 1)  
        ExitLong ("Condición 1 Entrada");  
  
    //Cierra la posición creada por la condición de entrada 2 si (CrossBelow (RSI  
    (14, 3), 70, 1))  
        ExitLong ("Condición 2 Entrada");  
}
```

El Pedido de Internal que Maneja Reglas que Reducen Posiciones No deseadas

NinjaTrader es una plataforma comercial viva de tiempo real y así tenemos que asegurar que prevenimos situaciones en el de tiempo real donde usted puede tener funcionamiento de pedidos múltiple llevando a cabo la misma tarea. Un ejemplo de esto sería si su estrategia tuviera un orden límite objetivo para 1 funcionamiento de contrato y luego su estrategia publicó un orden al precio de mercado para cerrar la posición. ¡Esto es peligroso ya que esto podría causar ambos pedidos lleno y en vez de tener una posición llana usted es corto ahora! Para prevenir estos tipos de situaciones, hay unos "bajo la capucha" dictaminan que una estrategia NinjaScript sigue cuando los métodos de pedido son llamados.

En su mayor parte, usted no tiene que ser íntimamente familiar con estas reglas ya que usted desarrolla sus estrategias, su todos tenidos cuidado de para usted interno dentro de una estrategia. Si una regla es violada, usted será notificado por un registro de errores en la etiqueta de tronco de Centro de Control dirigiendo una estrategia en el de tiempo real o en un backtest.

Las reglas siguientes son verdad por órdenes al precio de mercado de exclusión de nombre de señal únicos:

Los métodos que generan pedidos (excluyendo órdenes al precio de mercado) para **entrar** en una posición no serán ignorados si: el • Una posición está abierto y un pedido presentado por un método de salida ([ExitLongLimit \(\)](#) por ejemplo)

es activo y el pedido es usado para abrir una posición en dirección contraria • Una posición está abierta y un pedido presentado por un método de juego ([SetStopLoss \(\)](#) por ejemplo) es

activo y el pedido es usado para abrir una posición en dirección contraria • La posición de estrategia es llana y un pedido presentado por un método entrar ([EnterLongLimit \(\)](#) por ejemplo) es activo y el pedido es usado para abrir una posición en dirección contraria

Los métodos que generan pedidos de **salir** de una posición no serán ignorados si una posición de estrategia está abierta y: el • Un pedido presentado por un método entrar ([EnterLongLimit \(\)](#) por ejemplo) es activo y esto

el pedido de entrada es usado para abrir una posición en dirección contraria • Un pedido presentado por un método de juego ([SetStopLoss \(\)](#) por ejemplo) es activo

Juego () los métodos que generan pedidos de **salir** de una posición no serán ignorados si una posición de estrategia es

p293

www.ninjatrader.com

© NinjaTrader, LLC,

2005

abierto y:

• Un pedido presentado por un método entrar ([EnterLongLimit \(\)](#) por ejemplo) es activo y esto

el pedido de entrada es usado para abrir una posición en dirección contraria • Un pedido presentado por un método de salida ([ExitLongLimit \(\)](#) por ejemplo) es activo

Manejo de Pedido avanzado

El manejo de pedido avanzado es reservado para programadores **CON EXPERIENCIA**. Por el pedido avanzado que le maneja puede presentar, cambiar y anular pedidos en su discreción por cualquier [método conducido por el acontecimiento](#) dentro de una estrategia. Cada método de pedido en la "sección" de Métodos de Pedido hace diseñar una sobrecarga de método para el manejo avanzado.

Vivo Hasta que los Pedidos **de Pedidos Anulados** pueden ser presentados que son vivos hasta que usted llame [CancelOrder \(\)](#) el método o el tiempo del pedido vigente ha expirado, que alguna vez viene primero. Esta flexibilidad permite que usted controle exactamente cuando un pedido debería ser anulado en vez de confiar en el final de una barra. Cada método de pedido, como [EnterLongLimit \(\)](#) tiene una variación de método (sobrecarga) diseñada para presentar un vivo hasta el pedido anulado.

La Clase IOrder Todos los métodos de pedido devuelve un objeto [de IOrder](#). Varias cosas usted tiene que ser consciente de:

- Un objeto de IOrder volvió de llamar un método de pedido es dinámico en aquel las propiedades de los it
siempre reflejará que el estado actual de un pedido • Un objeto de IOrder pasó en de [OnOrderUpdate \(\)](#) el método es volátil en esto los its'
las propiedades permanecen estáticas y no cambian. No agarre a estas referencias para la comprobación de las propiedades de los it fuera de OnOrderUpdate () el método.
Compruebe que el objeto de IOrder volvió de la llamada original al método de pedido.
- Los objetos de IOrder no son **únicos** por pedido y por lo tanto no comparan objetos de IOrder con
el uno al otro para determinar la igualdad • Para comprobar la igualdad comprueban <IOrder>.Token la propiedad que es un único garantizado
el identificador para un pedido • La propiedad <IOrder>.OrderId no es **un** valor único ya que esto puede cambiar en todas partes un
la vida del pedido

El ejemplo de código siguiente demuestra que la sumisión de un pedido y la asignación de la vuelta de IOrder se oponen a la variable "entryOrder" y luego comprobación de la propiedad "Simbólica" en OnOrderUpdate () método para igualdad y luego comprobación si el pedido estuviera lleno.

```

IOrder privado entryOrder = nulo;

protegido anulan OnBarUpdate vacío () {
    si (entryOrder == nulo && Cerca [0]> Abierto [0])
        entryOrder = EnterLong ();}

protegido anulan OnOrderUpdate vacío (pedido de IOrder) {
    si (!entryOrder! = nulo && entryOrder. Señal == pedido.
        Señal) {
        Letra (pedido. ToString ()); si (pedido. OrderState ==
            OrderState. Lleno)
            entryOrder = nulo;
    }}

```

El funcionamiento con una Estrategia de Multiinstrumento Con el pedido avanzado que le maneja puede presentar unos pedidos en el contexto de cualquier objeto de barras designando el índice "BarsInProgress". Lo que esto significa es que si su serie de barra primaria es "MSFT" y su serie secundaria añadida a la estrategia a través de [Añadir \(\)](#) el método es "AAPL", usted puede presentar un pedido por "MSFT" o por "AAPL" en todas partes desde dentro la estrategia. Hay información detallada del funcionamiento con [multimarca de tiempo y estrategias de instrumento](#) sin embargo, esta subdivisión estrictamente trata con el concepto de la sumisión de pedido.

Vaya a tomar como un ejemplo EnterLongLimit () método y una de sus variaciones de método (debajo) de el diseñado para el manejo de pedido avanzado:

EnterLongLimit (intervalo *barsInProgressIndex*, bool *liveUntilCancelled*, cantidad internacional, doble *limitPrice*, cuerda *signalName*)

También vaya a seguir con el ejemplo encima donde un "MSFT" carta de 1 minuto es la serie de barra primaria la estrategia corre en. Añadimos una serie de barra secundaria, un "AAPL" 1 serie de minuto a nuestra estrategia llamando Añadir () método en [Inicializar \(\)](#) método como sigue:

```
Añada ("AAPL", PeriodType. Minuto, 1);
```

En NinjaScript, "MSFT" tiene un valor de índice de barras de "0" y "AAPL" tendría el valor de índice de barras de "1". Más información sobre este concepto puede ser encontrada en la [multimarca de tiempo y sección de estrategias de instrumento](#) así como la propiedad [BarsInProgress](#). En una estrategia de multiinstrumento, usando EnterLongLimit () el método

encima generará un pedido por el instrumento que tiene que ver al valor del parámetro "barsInProgressIndex" en el cual usted pasa.

El ejemplo de código siguiente demuestra como usted puede supervisar para acontecimientos de actualización de barra en "MSFT", pero presentar un pedido a "AAPL". La llave está en el paso en un valor de "1" (rojo valiente abajo) en para el "BarsInProgressIndex". Un valor de 1 representa la serie de barra "AAPL" y así el pedido es presentado para "AAPL".

p295

www.ninjatrader.co

m

© NinjaTrader, LLC, 2005

```
IOrder privado entryOrder = nulo;

protegido anulan OnBarUpdate vacío () {
    //Compruebe si la serie MSFT provocó un acontecimiento de actualización de barra
    si (BarsInProgress == 0) {
        //Presente un pedido por APPL en el contexto del acontecimiento de
        actualización de barra MSFT si (entryOrder == nulo)
        entryOrder = EnterLongLimit (1, verdadero, 1, Bajo [1] [0], "Pedido
        de AAPL");
    }
}
```

-o-

CancelOrder ()

La definición Anula el pedido especificado.

- Este método es puesto a disposición para programadores con experiencia esto totalmente entendimiento el los conceptos del pedido avanzado [que se maneja](#) • Este método envían una solicitud anular al agente de bolsa y no garantizan que un pedido es completamente anulado. La mayor parte del tiempo usted puede esperar su pedido de volver el 100 % anulado. Un pedido puede estar completamente lleno o la parte rellenó el tiempo que usted envía la solicitud anular y el tiempo el cambio recibe la solicitud. Compruebe [OnOrderUpdate \(\)](#) método para el estado de un pedido que usted intentó al anulado.

Sintaxis [CancelOrder](#) (*pedido de IOrder*)

Parámetros

pedido Un objeto [de IOrder](#) que representa el pedido usted desea anular.

Ejemplos

```
IOrder privado myEntryOrder = nulo; intervalo privado barNumberOfOrder = 0;

protegido anulan OnBarUpdate vacío () {
    //Presente un pedido de entrada en el nivel más bajo de una barra si
    (myEntryOrder == nulo) {
        myEntryOrder = EnterLongLimit (0, verdadero, 1, Bajo [0], "Entrada
        Larga"); barNumberOfOrder = CurrentBar;
    }

    //Si más de 5 barras han pasado, anule el pedido de entrada si
    (;myEntryOrder! = nulo && CurrentBar> barNumberOfOrder + 5)
        CancelOrder (myEntryOrder);
}
```

-0-

EnterLong ()

La definición Genera un orden al precio de mercado de compra para entrar en una posición larga.

p296

www.ninjatrader.com

© NinjaTrader, LLC,

2005

Valor de retorno de método

Un objeto sólo para leer [IOrder](#) que representa el pedido. Reservado para programadores con experiencia, la información adicional puede ser encontrada dentro del [Pedido Avanzado que Maneja](#) la sección.

Sintaxis EnterLong () [EnterLong \(ensartan signalName\)](#) [EnterLong \(cantidad internacional\)](#)
[EnterLong \(cantidad internacional, ensarte signalName\)](#)

La variación de método siguiente es para programadores con experiencia que totalmente entienden [el Pedido Avanzado que Maneja](#) conceptos.

[EnterLong \(intervalo los barsInProgressIndex, cantidad internacional, ensartan signalName\)](#)

Parámetros

cantidad de signalName barsInProgressInd
excepto

Ejemplos

El usuario definió el nombre de señal que identifica el pedido generado. Cantidad de pedido

de entrada. El índice de las Barras objeto a que el pedido fuera ser presentado contra. Ver la propiedad [de BarsInProgress](#). Esto determina para que instrumento el pedido es presentado.

```
protegido anulan OnBarUpdate vacío () {  
    si (CurrentBar <20)  
        vuelva;  
  
    //Sólo entre si al menos 10 barras han pasado desde nuestra última entrada si  
    (BarsSinceEntry ()> 10 && CrossAbove (SMA (10), SMA (20), 1))  
        EnterLong ("Entrada de Cruz de SMA");}
```

Las puntas (también ven [la Descripción](#)) • usando una firma de método que no tiene la cantidad de parámetro, la cantidad de pedido será tomado del conjunto de valores de cantidad en la ventana de diálogo de estrategia corriendo o backtesting una estrategia

o-

EnterLongLimit ()

Definición

Genera un orden límite de compra para entrar en una posición larga.

El Valor de retorno de método Un objeto sólo para leer [IOrder](#) que representa el pedido. Reservado para programadores con experiencia, la información adicional puede ser encontrada dentro del [Pedido Avanzado que Maneja](#) la sección.

EnterLongLimit de sintaxis (doblan *limitPrice*) EnterLongLimit (doblan *limitPrice*, *signalName* de cuerda) EnterLongLimit (*cantidad internacional*, *doble limitPrice*)

p297

www.ninjatrader.com
© NinjaTrader, LLC, 2005

EnterLongLimit (*cantidad internacional*, *doble limitPrice*, *ensarte signalName*)

La variación de método siguiente es para programadores con experiencia que totalmente entienden [el Pedido Avanzado que Maneja](#) conceptos.

EnterLongLimit (*intervalo barsInProgressIndex*, *bool los liveUntilCancelled*, *cantidad internacional*, *doblan limitPrice*, *ensartan signalName*)

Parámetros

signalName El usuario definió el nombre de señal que identifica el pedido generado. **limitPrice** El precio de límite del pedido. **cantidad** Cantidad de

pedido de entrada.

el `liveUntilCancelled` El pedido **no expirará** al final de barra, pero en cambio permanecerá vivo

hasta [CancelOrder \(\)](#) el método es llamado o su tiempo vigente ha sido alcanzado.

el `barsInProgressInd` El índice de las Barras objeto el pedido debe ser presentado contra. Ver

excepto la propiedad [de BarsInProgress](#). Esto determina que instrumento el pedido es presentado para.

Ejemplos

```
protegido anulan OnBarUpdate vacío () {
    si (CurrentBar <20)
        vuelva;

    //Sólo entre si al menos 10 barras han pasado desde nuestra última entrada si
    (BarsSinceEntry () > 10 && CrossAbove (SMA (10), SMA (20), 1))
        EnterLongLimit (GetCurrentBid (), "Entrada de Cruz de SMA");}
```

Las puntas (también ven la Descripción) • usando una firma de método que no tiene la cantidad de parámetro, la cantidad de pedido

será tomado del conjunto de valores de cantidad en la ventana de diálogo de estrategia corriendo o backtesting una estrategia

-O-

EnterLongStop ()

La definición Genera un pedido de parada de compra de entrar en una posición larga.

El Valor de retorno de método Un objeto sólo para leer [IOrder](#) que representa el pedido. Reservado para programadores con experiencia, la información adicional puede ser encontrada dentro del [Pedido Avanzado que Maneja](#) la sección.

EnterLongStop de sintaxis (`doblan stopPrice`) [EnterLongStop \(doblan stopPrice, signalName de cuerda\)](#) [EnterLongStop \(cantidad internacional, doble stopPrice\)](#) [EnterLongStop \(la cantidad internacional, doble stopPrice, ensarte signalName\)](#)

La variación de método siguiente es para programadores con experiencia que totalmente entienden [el Pedido Avanzado que Maneja](#) conceptos.

[EnterLongStop \(intervalo barsInProgressIndex, bool los liveUntilCancelled, cantidad internacional, doblan stopPrice\)](#)

p298

www.ninjatrader.com
© NinjaTrader, LLC,

2005

, [ensarte signalName](#))

Parámetro

S

signalName El usuario definió el nombre de señal que identifica el pedido generado. stopPrice El precio de parada del pedido. cantidad Cantidad de pedido de entrada. el liveUntilCancelled El pedido **no expirará** al final de barra, pero en cambio permanecerá vivo hasta [CancelOrder \(\)](#) el método es llamado o su tiempo vigente ha sido alcanzado. el barsInProgressInd El índice de las Barras objeto el pedido debe ser presentado contra. Ver excepto la propiedad [de BarsInProgress](#). Esto determina que instrumento el pedido es presentado para.

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    si (CurrentBar <20)  
        vuelva;  
  
    //Sólo entre si al menos 10 barras han pasado desde nuestra última entrada si  
    (BarsSinceEntry () > 10 && CrossAbove (SMA (10), SMA (20), 1))  
        EnterLongStop (GetCurrentAsk () + TickSize, "Entrada de Cruz de SMA");}
```

Las puntas (también ven [la Descripción](#)) • usando una firma de método que no tiene la cantidad de parámetro, la cantidad de pedido será tomado del conjunto de valores de cantidad en la ventana de diálogo de estrategia corriendo o backtesting una estrategia

-0-

EnterLongStopLimit ()

La definición Genera un orden límite de parada de compra para entrar en una posición larga.

El Valor de retorno de método Un objeto sólo para leer [IOrder](#) que representa el pedido. Reservado para programadores con experiencia, la información adicional puede ser encontrada dentro del [Pedido Avanzado que Maneja](#) la sección.

EnterLongStopLimit de sintaxis (doblan limitPrice, doblan stopPrice) EnterLongStopLimit (doblan limitPrice, doble stopPrice, signalName de cuerda) EnterLongStopLimit (cantidad internacional, doble limitPrice, doble stopPrice) EnterLongStopLimit (la cantidad internacional, doble limitPrice, doble stopPrice, ensarte signalName)

La variación de método siguiente es para programadores con experiencia que totalmente entienden [el Pedido Avanzado que Maneja](#) conceptos.

EnterLongStopLimit (intervalo barsInProgressIndex, bool los liveUntilCancelled, cantidad internacional, doblan limitPrice, doblan stopPrice, ensartan signalName)

Parámetros

signalName El usuario definió el nombre de señal que identifica el pedido generado.
limitPrice El precio de límite del pedido.

p299

www.ninjatrader.com
© NinjaTrader, LLC,

2005

stopPrice El precio de límite del pedido. cantidad
Cantidad de pedido de entrada.
liveUntilCancelled El pedido **no expirará** al final de barra, pero en cambio permanecerá vivo hasta [CancelOrder \(\)](#) el método es llamado o su tiempo vigente ha sido alcanzado.
barsInProgressInd El índice de las Barras objeto el pedido debe ser presentado contra. Ver
excepto la propiedad [de BarsInProgress](#). Esto determina que instrumento el pedido es presentado para.

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    si (CurrentBar <20)  
        vuelva;  
  
    //Sólo entre si al menos 10 barras han pasado desde nuestra última entrada si  
    (BarsSinceEntry () > 10 && CrossAbove (SMA (10), SMA (20), 1))  
        EnterLongStop (Alto [0] + 2 * TickSize, Alto [0], "Entrada de Cruz de SMA"); }
```

Las puntas (también ven [la Descripción](#)) • usando una firma de método que no tiene la cantidad de parámetro, la cantidad de pedido será tomado del conjunto de valores de cantidad en la ventana de diálogo de estrategia corriendo o backtesting una estrategia

-O-

EnterShort ()

La definición Genera vender el orden al precio de mercado corto para entrar en una posición corta.

El Valor de retorno de método Un objeto sólo para leer [IOrder](#) que representa el pedido. Reservado para programadores con experiencia, la información adicional puede ser encontrada dentro del [Pedido Avanzado que Maneja](#) la sección.

Sintaxis EnterShort () [EnterShort \(ensartan signalName\)](#) [EnterShort \(cantidad internacional\)](#)
[EnterShort \(cantidad internacional, ensarte signalName\)](#)

La variación de método siguiente es para programadores con experiencia que totalmente entienden [el Pedido Avanzado que Maneja](#) conceptos.

EnterShort (intervalo los barsInProgressIndex, cantidad internacional, ensartan signalName)

Parámetros

cantidad de signalName barsInProgressInd excepto

Ejemplos

El usuario definió el nombre de señal que identifica el pedido generado. Cantidad de pedido de entrada. El índice de las Barras objeta a que el pedido fuera ser presentado contra. Ver la propiedad [de BarsInProgress](#). Esto determina que instrumento el pedido es presentado para.

p300
www.ninjatrader.com
© NinjaTrader, LLC,
2005

```
protegido anulan OnBarUpdate vacío () {  
    si (CurrentBar <20)  
        vuelva;  
  
    //Sólo entre si al menos 10 barras han pasado desde nuestra última  
    entrada si (BarsSinceEntry () > 10 && CrossBelow (SMA (10), SMA (20),  
    1))  
        EnterShort ("Entrada de Cruz de SMA");}
```

Las puntas (también ven [la Descripción](#)) • usando una firma de método que no tiene la cantidad de parámetro, la cantidad de pedido será tomado del conjunto de valores de cantidad en la ventana de diálogo de estrategia corriendo o backtesting una estrategia

-0-

EnterShortLimit ()

La definición Genera vender el orden límite corto para entrar en una posición corta.

El Valor de retorno de método Un objeto sólo para leer [IOrder](#) que representa el pedido. Reservado para programadores con experiencia, la información adicional puede ser encontrada dentro del [Pedido Avanzado que Maneja](#) la sección.

EnterShortLimit de sintaxis (doblan limitPrice) **EnterShortLimit** (doblan limitPrice, signalName de cuerda) **EnterShortLimit** (cantidad internacional, doble limitPrice) **EnterShortLimit** (la cantidad internacional, doble limitPrice, ensarte signalName)

La variación de método siguiente es para programadores con experiencia que totalmente entienden [el Pedido Avanzado que Maneja](#) conceptos.

[EnterShortLimit](#) ([intervalo](#) [barsInProgressIndex](#), [bool](#) [los liveUntilCancelled](#), [cantidad](#) [internacional](#), [doblan](#) [limitPrice](#), [ensartan](#) [signalName](#))

Parámetros

[signalName](#) [limitPrice](#) [cantidad](#) [liveUntilCancelled](#)

El usuario definió el nombre de señal que identifica el pedido generado. El precio de límite del pedido. Cantidad de pedido de entrada. El pedido **no expirará** al final de barra, pero en cambio permanecerá vivo hasta [CancelOrder \(\)](#) el método es llamado o su tiempo vigente ha sido alcanzado.

[el barsInProgressInd](#) El índice de las Barras objeto el pedido debe ser presentado contra. Ver

excepto la propiedad [de BarsInProgress](#). Esto determina que instrumento el pedido es presentado para.

Ejemplos

p301

[www.ninjatrader.co](#)

m

© NinjaTrader, LLC,
2005

```
protegido anulan OnBarUpdate vacío () {  
    si (CurrentBar <20)  
        vuelva;  
  
    //Sólo entre si al menos 10 barras han pasado desde nuestra última  
    entrada si (BarsSinceEntry () > 10 && CrossBelow (SMA (10), SMA (20),  
    1))  
        EnterShortLimit (GetCurrentAsk (), "Entrada de Cruz de SMA");}
```

Las puntas (también ven [la Descripción](#)) • usando una firma de método que no tiene la cantidad de parámetro, la cantidad de pedido

será tomado del conjunto de valores de cantidad en la ventana de diálogo de estrategia corriendo o
backtesting una estrategia

EnterShortStop ()

La definición Genera vender el pedido de parada corto de entrar en una posición corta.

El Valor de retorno de método Un objeto sólo para leer [IOrder](#) que representa el pedido. Reservado para programadores con experiencia, la información adicional puede ser encontrada dentro del [Pedido Avanzado que Maneja](#) la sección.

EnterShortStop de sintaxis (doblan stopPrice) EnterShortStop (doblan stopPrice, signalName de cuerda) EnterShortStop (cantidad internacional, doble stopPrice) EnterShortStop (la cantidad internacional, doble stopPrice, ensarte signalName)

La variación de método siguiente es para programadores con experiencia que totalmente entienden el [Pedido Avanzado que Maneja](#) conceptos.

EnterShortStop (intervalo barsInProgressIndex, bool los liveUntilCancelled, cantidad internacional, doblan stopPrice, ensartan signalName)

Parámetros

signalName stopPrice cantidad liveUntilCancelled

El usuario definió el nombre de señal que identifica el pedido generado. El precio de parada del pedido. Cantidad de pedido de entrada. El pedido **no expirará** al final de barra, pero en cambio permanecerá vivo hasta [CancelOrder \(\)](#) el método es llamado o su tiempo vigente ha sido alcanzado. el barsInProgressInd El índice de las Barras objeta el pedido debe ser presentado contra. Ver exceptola propiedad [de BarsInProgress](#). Esto determina que instrumento el pedido es presentado para.

Ejemplos

p302

www.ninjatrader.co

m

© NinjaTrader, LLC,
2005

```
protegido anulan OnBarUpdate vacío () {
    si (CurrentBar <20)
        vuelva;

    //Sólo entre si al menos 10 barras han pasado desde nuestra última
    entrada si (BarsSinceEntry () > 10 && CrossBelow (SMA (10), SMA (20),
    1))
        EnterShortStop (GetCurrentBid () + TickSize, "Entrada de Cruz de
```

```
SMA") ; }
```

Las puntas (también ven [la Descripción](#)) • usando una firma de método que no tiene la cantidad de parámetro, la cantidad de pedido será tomado del conjunto de valores de cantidad en la ventana de diálogo de estrategia corriendo o backtesting una estrategia

-o-

EnterShortStopLimit ()

La definición Genera vender el orden límite de parada corto para entrar en una posición corta.

El Valor de retorno de método Un objeto sólo para leer [IOrder](#) que representa el pedido. Reservado para programadores con experiencia, la información adicional puede ser encontrada dentro del [Pedido Avanzado que Maneja](#) la sección.

EnterShortStopLimit de sintaxis (doblan *limitPrice*, doblan *stopPrice*) EnterShortStopLimit (doblan *limitPrice*, doble *stopPrice*, *signalName* de cuerda) EnterShortStopLimit (*cantidad internacional*, doble *limitPrice*, doble *stopPrice*) EnterShortStopLimit (*la cantidad internacional*, doble *limitPrice*, doble *stopPrice*, ensarte *signalName*)

La variación de método siguiente es para programadores con experiencia que totalmente entienden el [Pedido Avanzado que Maneja](#) conceptos.

EnterShortStopLimit (*intervalo barsInProgressIndex*, bool *los liveUntilCancelled*, *cantidad internacional*, doblan *limitPrice*, doblan *stopPrice*, ensartan *signalName*)

Parámetros

signalName limitPrice stopPrice cantidad liveUntilCancelled

El usuario definió el nombre de señal que identifica el pedido generado. El precio de límite del pedido. El precio de límite del pedido. Cantidad de pedido de entrada. El pedido **no expirará** al final de barra, pero en cambio permanecerá vivo hasta [CancelOrder \(\)](#) el método es llamado o su tiempo vigente ha sido alcanzado.

el *barsInProgressInd* El índice de las Barras objeta el pedido debe ser presentado contra.

Ver

exceptola propiedad [de BarsInProgress](#). Esto determina que instrumento el pedido es presentado para.

Ejemplos

www.ninjatrader.co

m
© NinjaTrader, LLC,
2005

```
protegido anulan OnBarUpdate vacío () {  
    si (CurrentBar <20)  
        vuelva;  
  
    //Sólo entre si al menos 10 barras han pasado desde nuestra última entrada  
    si (BarsSinceEntry () > 10 && CrossBelow (SMA (10), SMA (20), 1))  
        EnterShortStop (Bajo [0] + 2 * TickSize, Bajo [0], "Entrada de Cruz  
de SMA");}
```

Las puntas (también ven [la Descripción](#)) • usando una firma de método que no tiene la cantidad de parámetro, la cantidad de pedido será tomado del conjunto de valores de cantidad en la ventana de diálogo de estrategia corriendo o backtesting una estrategia

-0-

ExitLong ()

La definición Genera un orden al precio de mercado vender para salir de una posición larga.

El Valor de retorno de método Un objeto sólo para leer [IOrder](#) que representa el pedido. Reservado para programadores con experiencia, la información adicional puede ser encontrada dentro del [Pedido Avanzado que Maneja](#) la sección.

ExitLong **de sintaxis** () [ExitLong \(cantidad internacional\)](#) [ExitLong \(ensartan fromEntrySignal\)](#) [ExitLong \(ensartan signalName, fromEntrySignal de cuerda\)](#) [ExitLong \(cantidad internacional, signalName de cuerda, fromEntrySignal de cuerda\)](#)

La variación de método siguiente es para programadores con experiencia que totalmente entienden [el Pedido Avanzado que Maneja](#) conceptos.

[ExitLong \(intervalo los barsInProgressIndex, cantidad internacional, ensartan signalName, ensartan fromEntrySignal\)](#)

Parámetros

signalName El usuario definió el nombre de señal que identifica el pedido generado.

fromEntrySignal El nombre de señal de entrada. Esto ata la salida a la entrada y sale el

la cantidad de posición representada por la entrada actual.

cantidad Cantidad de pedido de entrada.

el barsInProgressInd El índice de las Barras objeto el pedido debe ser presentado contra.

Ver

exceptola propiedad [de BarsInProgress](#). Esto determina que instrumento el pedido es presentado para.

Ejemplos

p304

www.ninjatrader.co

m

© NinjaTrader, LLC,
2005

```
protegido anulan OnBarUpdate vacío () {  
    si (CurrentBar <20)  
        vuelva;  
  
    //Sólo entre si al menos 10 barras han pasado desde nuestra última  
    entrada si (BarsSinceEntry () > 10 && CrossAbove (SMA (10), SMA (20),  
    1))  
        EnterLong ("Entrada de Cruz de SMA");  
  
    //Posición de salidas si (CrossBelow (SMA (10), SMA (20), 1))  
        ExitLong ();}
```

Las puntas (también ven la Descripción) • Este método no son ignoradas si una posición larga no existe • es provechoso proporcionar un nombre de señal si su estrategia tiene puntos de salida múltiples para ayudar

identifique sus salidas en una carta • Usted puede atar una salida a una entrada proporcionando el nombre de señal de entrada en el parámetro

salen "del fromEntrySignal" • Si usted no especifica una cantidad la posición entera dando su piso de estrategia • Si usted no especifica un parámetro "fromEntrySignal" salen de la posición entera dando su piso de estrategia

-0-

ExitLongLimit ()

La definición Genera un orden límite vender para salir de una posición larga.

El Valor de retorno de método Un objeto sólo para leer **IOrder** que representa el pedido. Reservado para programadores con experiencia, la información adicional puede ser encontrada dentro del [Pedido Avanzado que Maneja](#) la sección.

Sintaxis ExitLongLimit (doblan limitPrice) ExitLongLimit (cantidad internacional, doble limitPrice) ExitLongLimit (doble limitPrice, ensarte fromEntrySignal) ExitLongLimit (doble limitPrice, ensarte signalName, ensarte fromEntrySignal) ExitLongLimit (cantidad internacional, doble limitPrice, ensarte signalName, ensarte fromEntrySignal)

La variación de método siguiente es para programadores con experiencia que totalmente entienden [el Pedido Avanzado que Maneja](#) conceptos.

ExitLongLimit (intervalo barsInProgressIndex, bool los liveUntilCancelled, cantidad internacional, doblan limitPrice, ensartan signalName, ensartan fromEntrySignal)

Parámetros

signalName fromEntrySignal

cantidad de limitPrice

El usuario definió el nombre de señal que identifica el pedido generado. El nombre de señal de entrada. Esto ata la salida a la entrada y sale de la cantidad de posición representada por la entrada actual. El precio de límite del pedido. Cantidad de pedido de entrada.

p305

www.ninjatrader.com

© NinjaTrader, LLC, 2005

el liveUntilCancelled El pedido **no expirará** al final de barra, pero en cambio permanecerá vivo

hasta [CancelOrder \(\)](#) el método es llamado o su tiempo vigente ha sido alcanzado.

el barsInProgressInd El índice de las Barras objeta el pedido debe ser presentado contra. Ver

exceptola propiedad [de BarsInProgress](#). Esto determina que instrumento el pedido es presentado para.

Ejemplos

```
protegido anulan OnBarUpdate vacío () {
    si (CurrentBar <20)
        vuelva;

    //Sólo entre si al menos 10 barras han pasado desde nuestra última entrada si
    (BarsSinceEntry () > 10 && CrossAbove (SMA (10), SMA (20), 1))
        EnterLong ("Entrada de Cruz de SMA");

    //Posición de salidas si (CrossBelow (SMA (10), SMA (20), 1))
        ExitLongLimit (GetCurrentBid);}
```

Las puntas (también ven la Descripción) • Este método no son ignoradas si una posición larga no existe • es provechoso proporcionar un nombre de señal si su estrategia

tiene puntos de salida múltiples para ayudar

identifique sus salidas en una carta • Usted puede atar una salida a una entrada proporcionando el nombre de señal de entrada en el parámetro

salen "del fromEntrySignal" • Si usted no especifica una cantidad la posición entera dando su piso de estrategia • Si usted no especifica un parámetro "fromEntrySignal" salen de la posición entera dando su piso de estrategia

-O-

ExitLongStop ()

La definición Genera un pedido de parada vender de salir de una posición larga.

El Valor de retorno de método Un objeto sólo para leer [IOrder](#) que representa el pedido. Reservado para programadores con experiencia, la información adicional puede ser encontrada dentro del [Pedido Avanzado que Maneja](#) la sección.

Sintaxis [ExitLongStop \(doblan stopPrice\)](#) [ExitLongStop \(cantidad internacional, doble stopPrice\)](#) [ExitLongStop \(doble stopPrice, ensarte fromEntrySignal\)](#) [ExitLongStop \(doble stopPrice, ensarte signalName, ensarte fromEntrySignal\)](#) [ExitLongStop \(cantidad internacional, doble stopPrice, ensarte signalName, ensarte fromEntrySignal\)](#)

La variación de método siguiente es para programadores con experiencia que totalmente entienden [el Pedido Avanzado que Maneja](#) conceptos.

[ExitLongStop \(intervalo barsInProgressIndex, bool los liveUntilCancelled, cantidad internacional, doblan stopPrice, ensartan signalName, ensartan fromEntrySignal\)](#)

p306

www.ninjatrader.com

signalName fromEntrySignal

cantidad de stopPrice

© NinjaTrader, LLC, 2005

El Usuario **de parámetros** definió el nombre de señal que identifica el pedido generado. El nombre de señal de entrada. Esto ata la salida a la entrada y sale el la cantidad de posición representada por la entrada actual. El precio de parada del pedido. Cantidad de pedido de entrada. el liveUntilCancelled El pedido **no expirará** al final de barra, pero en cambio permanecerá vivo

hasta [CancelOrder \(\)](#) el método es llamado o su tiempo vigente ha sido alcanzado.

el barsInProgressInd El índice de las Barras objeta el pedido debe ser presentado contra.

Ver

excepto la propiedad [de BarsInProgress](#). Esto determina que instrumento el pedido es presentado para.

Ejemplos

```
double stopPrice privado = 0;

protegido anulan OnBarUpdate vacío () {
    si (CurrentBar < 20)
        vuelva;

    //Sólo entre si al menos 10 barras han pasado desde nuestra última entrada si
    (BarsSinceEntry () > 10 && CrossAbove (SMA (10), SMA (20), 1)) {
        EnterLong ("Entrada de Cruz de SMA"); stopPrice = Bajo [0];
    }

    //Posición de salidas ExitLongStop (stopPrice);
}
```

Las puntas (también ven la Descripción) • Este método no son ignoradas si una posición larga no existe • es provechoso proporcionar un nombre de señal si su estrategia tiene puntos de salida múltiples para ayudar

identifique sus salidas en una carta • Usted puede atar una salida a una entrada proporcionando el nombre de señal de entrada en el parámetro

salen "del fromEntrySignal" • Si usted no especifica una cantidad la posición entera dando su piso de estrategia • Si usted no especifica un parámetro "fromEntrySignal" salen de la posición entera dando

su piso de estrategia

-o-

ExitLongStopLimit ()

La definición Genera un orden límite de parada vender para salir de una posición larga.

El Valor de retorno de método Un objeto sólo para leer [IOrder](#) que representa el pedido. Reservado para programadores con experiencia, la información adicional puede ser encontrada dentro del [Pedido Avanzado que Maneja](#) la sección.

ExitLongStopLimit de sintaxis (doblan *limitPrice*, doble *stopPrice*) [ExitLongStopLimit](#) (*cantidad internacional*, doble *limitPrice*, doble *stopPrice*)

p307

www.ninjatrader.com
© NinjaTrader, LLC, 2005

[ExitLongStopLimit](#) (doblan *limitPrice*, doblan *stopPrice*, ensartan *fromEntrySignal*)
[ExitLongStopLimit](#) (doblan *limitPrice*, doble *stopPrice*, *signalName* de cuerda, *fromEntrySignal* de cuerda) [ExitLongStopLimit](#) (*cantidad internacional*, dobles *limitPrice*, doble *stopPrice*, ensartan *signalName*, ensartan *fromEntrySignal*)

La variación de método siguiente es para programadores con experiencia que totalmente entienden [el Pedido Avanzado que Maneja](#) conceptos.

`ExitLongStopLimit` (`intervalo` `barsInProgressIndex`, `bool` `liveUntilCancelled`, `cantidad` `internacional`, `doblan` `limitPrice`, `doblan` `stopPrice`, `ensartan` `signalName`, `ensartan` `fromEntrySignal`)

Parámetros

`signalName` `fromEntrySignal`

`limitPrice` `stopPrice` `cantidad`

El usuario definió el nombre de señal que identifica el pedido generado. El nombre de señal de entrada. Esto ata la salida a la entrada y sale de la cantidad de posición representada por la entrada actual. El precio de límite del pedido El precio de parada del pedido. Cantidad de pedido de entrada.

el `liveUntilCancelled` El pedido **no expirará** al final de barra, pero en cambio permanecerá vivo

hasta `CancelOrder ()` el método es llamado o su tiempo vigente ha sido alcanzado.

el `barsInProgressInd` El índice de las Barras objeto el pedido debe ser presentado contra. Ver

excepto la propiedad [de BarsInProgress](#). Esto determina que instrumento el pedido es presentado para.

Ejemplos

```
double stopPrice privado = 0;
```

```
protegido anulan OnBarUpdate vacío () {  
    si (CurrentBar <20)  
        vuelva;
```

```
    //Sólo entre si al menos 10 barras han pasado desde nuestra última entrada si  
    (BarsSinceEntry () > 10 && CrossAbove (SMA (10), SMA (20), 1)) {  
        EnterLong ("Entrada de Cruz de SMA"); stopPrice = Bajo [0];  
    }
```

```
    //Posición de salidas ExitLongStopLimit (stopPrice - (10 * TickSize),  
    stopPrice);  
}
```

Las puntas (también ven [la Descripción](#)) • Este método no son ignoradas si una posición larga no existe • es provechoso proporcionar un nombre de señal si su estrategia tiene puntos de salida múltiples para ayudar

identifique sus salidas en una carta • Usted puede atar una salida a una entrada proporcionando el nombre de señal de entrada en el parámetro

salen "del `fromEntrySignal`" • Si usted no especifica una cantidad la posición entera dando su piso de estrategia • Si usted no especifica un parámetro "`fromEntrySignal`" salen de la posición entera dando

su piso de estrategia

-O-

p308

www.ninjatrader.com

© NinjaTrader, LLC,

2005

ExitShort ()

Definición

Genera una compra para cubrir el orden al precio de mercado para salir de una posición corta.

El Valor de retorno de método Un objeto sólo para leer [IOrder](#) que representa el pedido. Reservado para programadores con experiencia, la información adicional puede ser encontrada dentro del [Pedido Avanzado que Maneja](#) la sección.

ExitShort **de sintaxis** () ExitShort (cantidad internacional) ExitShort (ensartan fromEntrySignal) ExitShort (ensartan signalName, fromEntrySignal de cuerda) ExitShort (cantidad internacional, signalName de cuerda, fromEntrySignal de cuerda)

La variación de método siguiente es para programadores con experiencia que totalmente entienden [el Pedido Avanzado que Maneja](#) conceptos.

ExitShort (intervalo los barsInProgressIndex, cantidad internacional, ensartan signalName, ensartan fromEntrySignal)

Parámetros

signalName El usuario definió el nombre de señal que identifica el pedido generado.
fromEntrySignal El nombre de señal de entrada. Esto ata la salida a la entrada y sale el
la cantidad de posición representada por la entrada actual.
cantidad Cantidad de pedido de entrada.
el barsInProgressInd El índice de las Barras objeta el pedido debe ser presentado contra.
Ver
exceptola propiedad [de BarsInProgress](#). Esto determina que instrumento el pedido es presentado para.

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    si (CurrentBar <20)  
        vuelva;  
  
    //Sólo entre si al menos 10 barras han pasado desde nuestra última entrada si  
    (BarsSinceEntry () > 10 && CrossBelow (SMA (10), SMA (20), 1))  
        EnterShort ("Entrada de Cruz de SMA");  
  
    //Posición de salidas si (CrossAbove (SMA (10), SMA (20), 1))  
        ExitShort ();}
```

Las puntas (también ven [la Descripción](#)) • Este método no son ignoradas si una posición corta no existe • es provechoso proporcionar un nombre de señal si su estrategia tiene puntos de salida múltiples para ayudar

identifique sus salidas en una carta • Usted puede atar una salida a una entrada proporcionando el nombre de señal de entrada en el parámetro

salen "del fromEntrySignal" • Si usted no especifica una cantidad la posición entera dando su piso de estrategia • Si usted no especifica un parámetro "fromEntrySignal" salen de la posición entera dando su piso de estrategia

p309

www.ninjatrader.com
© NinjaTrader, LLC,

2005

-0-

ExitShortLimit ()

Definición

Genera una compra para cubrir el orden límite para salir de una posición corta.

El Valor de retorno de método Un objeto sólo para leer [IOrder](#) que representa el pedido. Reservado para programadores con experiencia, la información adicional puede ser encontrada dentro del [Pedido Avanzado que Maneja](#) la sección.

Sintaxis ExitShortLimit (doblan limitPrice) ExitShortLimit (cantidad internacional, doble limitPrice) ExitShortLimit (doble limitPrice, ensarte fromEntrySignal) ExitShortLimit (doble limitPrice, ensarte signalName, ensarte fromEntrySignal) ExitShortLimit (cantidad internacional, doble limitPrice, ensarte signalName, ensarte fromEntrySignal)

La variación de método siguiente es para programadores con experiencia que totalmente entienden [el Pedido Avanzado que Maneja](#) conceptos.

ExitShortLimit (intervalo barsInProgressIndex, bool los liveUntilCancelled, cantidad internacional, doblan limitPrice, ensarte signalName, ensarte fromEntrySignal)

Parámetros

signalName fromEntrySignal

cantidad de limitPrice

El usuario definió el nombre de señal que identifica el pedido generado. El nombre de señal de entrada. Esto ata la salida a la entrada y sale el la cantidad de posición representada por la entrada actual. El precio de límite del pedido. Cantidad de pedido de entrada.

el liveUntilCancelled El pedido **no expirará** al final de barra, pero en cambio permanecerá vivo

hasta [CancelOrder \(\)](#) el método es llamado o su tiempo vigente ha

sido alcanzado.
el barsInProgressInd El índice de las Barras objeta el pedido debe ser presentado contra. Ver
exceptola propiedad [de BarsInProgress](#). Esto determina que instrumento el pedido es presentado para.

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    si (CurrentBar <20)  
        vuelva;  
  
    //Sólo entre si al menos 10 barras han pasado desde nuestra última entrada  
    si (BarsSinceEntry () > 10 && CrossBelow (SMA (10), SMA (20), 1))  
        EnterShort ("Entrada de Cruz de SMA");  
  
    //Posición de salidas si (CrossAbove (SMA (10), SMA (20), 1))  
        ExitShortLimit (GetCurrentAsk);}
```

Las puntas (también ven [la Descripción](#)) • Este método no son ignoradas si una posición larga no existe • es provechoso proporcionar un nombre de señal si su estrategia tiene puntos de salida múltiples para ayudar
identifique sus salidas en una carta

p310

www.ninjatrader.com
© NinjaTrader, LLC, 2005

• Usted puede atar una salida a una entrada proporcionando el nombre de señal de entrada en el parámetro
salen "del fromEntrySignal" • Si usted no especifica una cantidad la posición entera dando su piso de estrategia • Si usted no especifica un parámetro "fromEntrySignal" salen de la posición entera dando su piso de estrategia

-O-

ExitShortStop ()

La definición Genera una compra para cubrir el pedido de parada de salir de una posición corta.

El Valor de retorno de método Un objeto sólo para leer [IOrder](#) que representa el pedido. Reservado para programadores con experiencia, la información adicional puede ser encontrada dentro del [Pedido Avanzado que Maneja](#) la sección.

Sintaxis [ExitShortStop \(doblan stopPrice\)](#) [ExitShortStop \(cantidad internacional, doble stopPrice\)](#) [ExitShortStop \(doble stopPrice, ensarte fromEntrySignal\)](#) [ExitShortStop \(doble stopPrice, ensarte signalName, ensarte fromEntrySignal\)](#) [ExitShortStop \(cantidad internacional, doble stopPrice, ensarte signalName, ensarte fromEntrySignal\)](#)

La variación de método siguiente es para programadores con experiencia que totalmente

entienden *el Pedido Avanzado que Maneja* conceptos.

ExitShortStop (intervalo *barsInProgressIndex*, *bool* los *liveUntilCancelled*, cantidad internacional, *doblan* *stopPrice*, *ensarte* *signalName*, *ensarte* *fromEntrySignal*)

Parámetros

signalName *fromEntrySignal*

cantidad de *stopPrice*

El usuario definió el nombre de señal que identifica el pedido generado. El nombre de señal de entrada. Esto ata la salida a la entrada y sale el la cantidad de posición representada por la entrada actual. El precio de parada del pedido. Cantidad de pedido de entrada.

el *liveUntilCancelled* El pedido **no expirará** al final de barra, pero en cambio permanecerá vivo

hasta [CancelOrder \(\)](#) el método es llamado o su tiempo vigente ha sido alcanzado.

el *barsInProgressInd* El índice de las Barras objeto el pedido debe ser presentado contra. Ver

excepto la propiedad [de BarsInProgress](#). Esto determina que instrumento el pedido es presentado para.

Ejemplos

p311

www.ninjatrader.co

m

© NinjaTrader, LLC,

2005

```
double stopPrice privado = 0;
```

```
protegido anulan OnBarUpdate vacío () {  
    si (CurrentBar <20)  
        vuelva;
```

```
//Sólo entre si al menos 10 barras han pasado desde nuestra última  
entrada si (BarsSinceEntry () > 10 && CrossBelow (SMA (10), SMA (20),
```

```

1)) {
    EnterShort ("Entrada de Cruz de SMA"); stopPrice = Alto [0];
}

//Posición de salidas ExitShortStop (stopPrice);
}

```

Las puntas (también ven la Descripción) • Este método no son ignoradas si una posición larga no existe • es provechoso proporcionar un nombre de señal si su estrategia tiene puntos de salida múltiples para ayudar

identifique sus salidas en una carta • Usted puede atar una salida a una entrada proporcionando el nombre de señal de entrada en el parámetro

salen "del fromEntrySignal" • Si usted no especifica una cantidad la posición entera dando su piso de estrategia • Si usted no especifica un parámetro "fromEntrySignal" salen de la posición entera dando

su piso de estrategia

-o-

ExitShortStopLimit ()

La definición Genera una compra para cubrir el orden límite de parada para salir de una posición corta.

El Valor de retorno de método Un objeto sólo para leer [IOrder](#) que representa el pedido. Reservado para programadores con experiencia, la información adicional puede ser encontrada dentro del [Pedido Avanzado que Maneja](#) la sección.

ExitShortStopLimit de sintaxis (doblan *limitPrice*, doble *stopPrice*) [ExitShortStopLimit](#) (*cantidad internacional*, doble *limitPrice*, doble *stopPrice*) [ExitShortStopLimit](#) (doble *limitPrice*, doble *stopPrice*, *ensarte fromEntrySignal*) [ExitShortStopLimit](#) (doble *limitPrice*, doble *stopPrice*, *ensarte signalName*, *ensarte fromEntrySignal*) [ExitShortStopLimit](#) (*la cantidad internacional*, doble *limitPrice*, doble *stopPrice*, *ensarte signalName*, *ensarte fromEntrySignal*)

La variación de método siguiente es para programadores con experiencia que totalmente entienden [el Pedido Avanzado que Maneja](#) conceptos.

[ExitShortStopLimit](#) (*intervalo barsInProgressIndex*, *bool los liveUntilCancelled*, *cantidad internacional*, *doblan limitPrice*, *doblan stopPrice*, *ensartan signalName*, *ensartan fromEntrySignal*)

Parámetros

signalName El usuario definió el nombre de señal que identifica el pedido generado.

fromEntrySignal

limitPrice stopPrice cantidad

© NinjaTrader, LLC, 2005

El nombre de señal de entrada. Esto ata la salida a la entrada y sale de la cantidad de posición representada por la entrada actual. El precio de límite del pedido El precio de parada del pedido. Cantidad de pedido de entrada.

el liveUntilCancelled El pedido **no expirará** al final de barra, pero en cambio permanecerá vivo

hasta [CancelOrder \(\)](#) el método es llamado o su tiempo vigente ha sido alcanzado.

el barsInProgressInd El índice de las Barras objeto el pedido debe ser presentado contra. Ver

excepto la propiedad [de BarsInProgress](#). Esto determina que instrumento el pedido es presentado para.

Ejemplos

```
double stopPrice privado = 0;
```

```
protegido anulan OnBarUpdate vacío () {  
    si (CurrentBar <20)  
        vuelva;  
  
    //Sólo entre si al menos 10 barras han pasado desde nuestra última entrada si  
    (BarsSinceEntry () > 10 && CrossBelow (SMA (10), SMA (20), 1)) {  
        EnterShort ("Entrada de Cruz de SMA"); stopPrice = Alto [0];  
    }  
  
    //Posición de salidas ExitShortStopLimit (stopPrice + (10 * TickSize),  
    stopPrice);  
}
```

Las puntas (también ven la Descripción) • Este método no son ignoradas si una posición larga no existe • es provechoso proporcionar un nombre de señal si su estrategia tiene puntos de salida múltiples para ayudar

identifique sus salidas en una carta • Usted puede atar una salida a una entrada proporcionando el nombre de señal de entrada en el parámetro

salen "del fromEntrySignal" • Si usted no especifica una cantidad la posición entera dando su piso de estrategia • Si usted no especifica un parámetro "fromEntrySignal" salen de la posición entera dando

su piso de estrategia

-0-

SetProfitTarget ()

La definición Genera un pedido objetivo de ganancia de límite de salir de una posición. Los pedidos objetivo de ganancia son verdaderas órdenes de trabajo presentadas inmediatamente al mercado para recibir una ejecución de un pedido de entrada.

- Se aconseja llamar este método desde dentro la estrategia `Initialize ()` método si su ganancia
 - el precio/compensación objetivo es
 - estático que Usted puede llamar este método desde dentro la estrategia `OnBarUpdate ()` el método le debería desear
 - cambiar dinámicamente el precio objetivo mientras en un puesto vacante el
 - Le debería llamar este método de cambiar dinámicamente el precio objetivo en la estrategia `OnBarUpdate ()` método, usted siempre debería reinicializar el valor de precio/compensación objetivo cuando su estrategia es llana por otra parte, el último conjunto de valores de precio/compensación será usado para generar su ganancia

p313

www.ninjatrader.com
© NinjaTrader, LLC,
2005

apunte el pedido de su siguiente puesto vacante

`SetProfitTarget` **de sintaxis** (*doble dinero*) `SetProfitTarget` (*modo de CalculationMode, doble valor*) `SetProfitTarget` (*ensartan fromEntrySignal, modo de CalculationMode, doble valor*)

Parámetros

modo monetario

valor

fromEntrySignal

Ejemplos

Pone la cantidad objetivo de ganancia en el dinero (ganancia de 500\$ por ejemplo) `CalculationMode`. Por ciento (0.01 = el 1 %) `CalculationMode`. Precio `CalculationMode`. Señala El valor el pedido objetivo de ganancia es compensado del precio de entrada de posición (la excepción usa el modo.Price donde 'el valor' representará el precio actual) El nombre de señal de entrada. Esto ata la salida objetivo de ganancia a la entrada y sale de la cantidad de posición representada por la entrada actual.

```
protegido anulan el vacío Inicializan () {
    //Sugiere que una orden 10 objetivo de ganancia señala lejos del precio de
    entrada avg SetProfitTarget (CalculationMode. Marcas, 10);
}
```

Las puntas (también ver [la Descripción](#))

- Los pedidos objetivo de ganancia son presentados en el de tiempo real en ejecuciones de entrada de
 - de pedidos de entrada Una estrategia generará o un pedido objetivo por cada

uno parcial se llenan de un pedido de entrada o un

el pedido por todos se llena. Ver la información adicional bajo la etiqueta [de Estrategias](#) de las Opciones dialogwindow.

- Si una [pérdida de parada](#) o el pedido [de parada de rastro](#) son generados además de un pedido objetivo de ganancia, ellos son presentado ya que OCO (uno anula otro) • Un pedido objetivo de ganancia es automáticamente anulado si la posición gerente está cerrada por otra estrategia generó el pedido de salida

-0-

SetStopLoss ()

La definición Genera un pedido de pérdida de parada de salir de una posición. Párese los pedidos de pérdida son verdaderas órdenes de trabajo (a menos que no simulado es especificado en cuyo caso el pedido de parada es en la localidad simulado y presentado como el mercado una vez provocado) presentado inmediatamente al mercado para recibir una ejecución de un pedido de entrada.

- Se aconseja llamar este método desde dentro la estrategia [Initialize \(\)](#) método si su parada el precio/compensación de pérdida es • estático que Usted puede llamar este método desde dentro la estrategia [OnBarUpdate \(\)](#) el método le debería desear cambiar dinámicamente el precio de pérdida de parada mientras en un puesto vacante el
- Le debería llamar este método de cambiar dinámicamente el precio de pérdida de parada en la estrategia [OnBarUpdate \(\)](#) método, usted siempre debería reinicializar el valor de precio/compensación de pérdida de parada cuando su estrategia es llana por otra parte, el último conjunto de valores de precio/compensación será usado para generar su pedido de pérdida de parada de su siguiente puesto vacante

p314

www.ninjatrader.com
© NinjaTrader, LLC,

2005

[SetStopLoss de sintaxis](#) ([doblan el dinero](#))

[SetStopLoss](#) ([doble dinero](#), [bool simulado](#)) [SetStopLoss](#) ([modo de CalculationMode](#), [doble valor](#)) [SetStopLoss](#) ([ensartan fromEntrySignal](#), [modo de CalculationMode](#), [doble valor](#), [bool simulado](#))

Parámetros

el dinero simuló el modo

valor

fromEntrySignal

Ejemplos

Pone la cantidad de pérdida de parada en el dinero (pérdida de 500\$ por ejemplo) De ser verdadero, simulará el pedido de parada y se rendirá ya que el mercado una vez provocó CalculationMode. Por ciento (0.01 = el 1 %) CalculationMode. Precio CalculationMode. Señala El valor el pedido de pérdida de parada es compensado del precio de entrada de posición (la excepción usa el modo.Price donde 'el valor' representará el precio actual) El nombre de señal de entrada. Esto ata la salida de pérdida de parada a la entrada y sale de la cantidad de posición representada por la entrada actual.

```
protegido anulan el vacío Inicializan () {  
    //Presenta una pérdida de parada de SetStopLoss de 500$ (500);  
}
```

Las puntas (también ver [la Descripción](#))

- SetStopLoss () método no puede **ser** usado simultáneamente con [SetTrailStop \(\)](#) método para la misma posición, si ambos métodos son llamados para la misma posición (fromEntrySignal) SetStopLoss () siempre tendrá prioridad. Usted puede sin embargo, usar ambos métodos en la misma estrategia si ellos se refieren a nombres de señal diferentes.
- Párese los pedidos de pérdida son presentados en el de tiempo real en ejecuciones de entrada de • de pedidos de entrada Una estrategia generará o un pedido de pérdida de parada por cada uno parcial se llenan de un pedido de entrada o un el pedido por todos se llena. Ver la información adicional bajo la etiqueta [de Estrategias](#) de las Opciones dialogwindow.
- Si un pedido [objetivo de ganancia](#) es generado además de un pedido de pérdida de parada, ellos son presentados como
OCO (uno anula otro) • Párese los pedidos de pérdida son presentados como órdenes al precio de mercado de parada • Un pedido de pérdida de parada es automáticamente anulado si la posición gerente está cerrada por el otro
la estrategia generó el pedido de salida

-0-

SetTrailStop ()

La definición Genera un pedido de parada de rastro de salir de una posición. Las paradas de rastro son ammended en una marca por la base de marca. Los pedidos de parada de rastro son verdaderas órdenes de trabajo (a menos que no simulado es especificado en cuyo caso el pedido de parada es en la localidad simulado y presentado como el mercado una vez provocado) presentado inmediatamente al mercado para recibir una ejecución de un pedido de entrada.

- Se aconseja llamar este método desde dentro la estrategia [Initialize \(\)](#) método si su rastro párese el precio/compensación es • estático que Usted puede llamar este método desde dentro la estrategia [OnBarUpdate \(\)](#) el método le debería desear

para cambiar dinámicamente el rastro para el precio mientras en un puesto vacante el

- Le debería llamar este método de cambiar dinámicamente el precio de parada de rastro en la estrategia

OnBarUpdate () método, usted siempre debería reinicializar el valor de precio/compensación de parada de rastro cuando su la estrategia es llana por otra parte, el último conjunto de valores de precio/compensación será usado para generar su pedido de parada de rastro de su siguiente puesto vacante

SetTrailStop de sintaxis (doble dinero) SetTrailStop (doble dinero, bool simulado)
SetTrailStop (modo de CalculationMode, doble valor) SetTrailStop (ensartan fromEntrySignal, modo de CalculationMode, doble valor, bool simulado)

Parámetros

el dinero simulado

modo

valor fromEntrySignal

Ejemplos

Pone la cantidad de parada de rastro en el dinero (ganancia de 500\$ por ejemplo) De ser verdadero, simulará la parada de rastro piden y se rinden como el mercado una vez CalculationMode provocado. Por ciento (0.01 = el 1 %) CalculationMode. Precio CalculationMode. Hace tictac La parada de rastro valoran El nombre de señal de entrada. Esto ata la salida de parada de rastro a la entrada y sale de la cantidad de posición representada por la entrada actual.

```
protegido anulan el vacio Inicializan () {  
    //Pone una parada de rastro de 12 marcas SetTrailStop (CalculationMode. Marcas,  
    12);  
}
```

Las puntas (también ver [la Descripción](#))

- SetTrailStop () método no puede **ser** usado simultáneamente con [SetStopLoss \(\)](#) método para la misma posición, si ambos métodos son llamados para la misma posición (fromEntrySignal) SetStopLoss () siempre tendrá prioridad. Usted puede sin embargo, usar ambos métodos en la misma estrategia si ellos se refieren a nombres de señal diferentes.
- Los pedidos de parada de rastro son presentados en el de tiempo real en ejecuciones de entrada de • de pedidos de entrada Una estrategia generará o un pedido de parada de rastro por cada uno parcial se llenan de un pedido de entrada o un el pedido por todos se llena. Ver la información adicional bajo la etiqueta [de Estrategias](#) de las Opciones dialogwindow.

- Si un pedido **objetivo de ganancia** es generado además de un pedido de parada de rastro, ellos son presentados como OCO (uno anula otro) • Los pedidos de parada de rastro son presentados como órdenes al precio de mercado de parada • Un pedido de parada de rastro es automáticamente anulado si la posición gerente está cerrada por el otro la estrategia generó el pedido de salida

-o-

Métodos de Rendimiento de estrategia y Propiedades

Rendimiento

Definición

p316
www.ninjatrader.com
© NinjaTrader, LLC, 2005

El rendimiento es el objeto paternal que sostiene todos los comercios y datos de rendimiento comerciales generados por una estrategia.

- Una estrategia NinjaScript puede generar ambos comercios sintéticos (comercios de backtest históricos) y los comercios de tiempo real ejecutados en una corriente de datos de tiempo real. Si usted desea tener acceso a comercios sólo de tiempo real, tenga acceso a la colección "RealtimeTrades"
- El primer comercio de la colección "RealtimeTrades" contendrá una ejecución de entrada sintética si la estrategia no era **llana** entonces usted comienza la estrategia.

Métodos y Propiedades

AllTrades

LongTrades

RealtimeTrades

ShortTrades

Ejemplos

Consigue un objeto de estrategia Gets a [TradeCollection](#) de objeto [de TradeCollection](#) el objeto de estrategia Gets a [TradeCollection](#) por la estrategia Gets a [TradeCollection](#) objeta la estrategia de todos los comercios generados por el

de comercios largos generados por

de comercios de tiempo real generados

de comercios cortos generados por

```
protegido anulan OnBarUpdate vacío () {  
    //La copia impresa del número de la Letra de comercios larga ("La estrategia  
    ha tomado" + Rendimiento. LongTrades. Conde + "mucho tiempo  
    comercios. ");}
```

o-

Clase comercial

Definición

Un Comercio es un completado se compran/venden o se venden/compran la transacción. Esto consiste en la ejecución de salida y una entrada.

Ejemplo 1: Compre 1 contrato ante un precio de 1000 y véndase 1 contrato ante un precio de 1001 es un comercio completo.

Ejemplo 2: Compre 2 contratos ante un precio de 1000 y venda el 1er contrato ante un precio de 1001, luego véndase el 2do contrato ante un precio de 1002 son **dos** comercios completados.

En el segundo ejemplo encima, dos objetos comerciales son creados para representar cada comercio individual. Cada objeto comercial sostendrá la **misma** ejecución de entrada para dos contratos ya que esta ejecución sola era la ejecución inicial para ambos comercios individuales.

Métodos y Propiedades

Comisión

Entrada

Un valor [internacional](#) que representa el número de comercios en la colección Consigue un objeto [de IExecution](#) que representa la entrada

p317

www.ninjatrader.com

Salida ProfitCurrency

ProfitPercent ProfitPoints

Cantidad ToString ()

Ejemplos

© NinjaTrader, LLC, 2005

Consigue un objeto [de IExecution](#) que representa la salida Una [doble](#) ganancia de representación de valor en el dinero por su parte unidad cambiada. Múltiple este valor por la Cantidad comercial para conseguir la ganancia total en dinero para el comercio. Una [doble](#) representación de valor saca ganancia como un porcentaje Una [doble](#) ganancia de representación de valor en puntos para una unidad cambiada. Multiplique este valor por la Cantidad comercial para conseguir la ganancia total en puntos para el comercio. Un valor [internacional](#) que representa la cantidad del comercio Una representación [de cuerda](#) del objeto Comercial

```
protegido anulan OnBarUpdate vacío () {  
    //Compruebe para asegurarse que hay al menos un comercio en la colección si  
    (Rendimiento. RealtimeTrades. Conde > 0) {  
        //Consiga el último Comercio comercial de tiempo real completado lastTrade  
        =  
Rendimiento. RealtimeTrades [Rendimiento. RealtimeTrades. Conde - 1];  
  
        //Cuenta PnL para el último comercio de tiempo real completado doblan  
lastProfit = lastTrade. ProfitCurrency * lastTrade. Cantidad;  
  
        //Pring PnL a la Letra de ventana Output ("La ganancia del último comercio  
es" + lastProfit);  
    }  
}
```

-0-

Clase de TradeCollection

Definición Una colección de objetos [Comerciales](#). Usted puede tener acceso a un objeto comercial proporcionando un valor de índice. El valor de índice es el sentido basado cero que el comercio más reciente tomado en una estrategia estará en un valor de índice del cero. El comercio menos reciente tomado estará en un valor de índice de los comercios totales en la colección menos 1.

Métodos y Propiedades

Conde

[GetTrades \(\)](#)

LosingTrades TradesPerformance WinningTrades

Ejemplos

Un valor **internacional** que representa el número de comercios en la colección Se pone un objeto de TradeCollection que representa una posición especificada Consigue un objeto de TradeCollection de perder comercios Se pone un objeto **de TradesPerformance** Consigue un objeto de TradeCollection de ganar comercios

p318

www.ninjatrader.com
© NinjaTrader, LLC, 2005

EJEMPLO 1

```
protegido anulan OnBarUpdate vacío () {  
    //Tiene acceso al primer/último comercio en la estrategia (el primer comercio  
    está en el índice 0)  
    //e imprime la ganancia como un porcentaje a la ventana Trade de salida  
    firstTrade = Rendimiento. AllTrades [0]; Cambie lastTrade = Rendimiento.  
    AllTrades [Rendimiento. AllTrades. Conde - 1];  
  
    si (;lastTrade! = nulo)  
        La letra ("La última ganancia comercial es" + lastTrade. ProfitPercent);  
  
    si (;firstTrade! = nulo)  
        La letra ("La primera ganancia comercial es" + firstTrade. ProfitPercent);}
```

```
EL EJEMPLO 2 protegido anula OnBarUpdate vacío () {  
    //Una vez que la estrategia ha ejecutado 20 lazo de comercios por los comercios  
    que pierden//colección y copia impresa de PnL en comercios sólo largos si  
    (Rendimiento. AllTrades. Conde == 20) {  
        Letra (Performance.AllTrades.LosingTrades.Count.ToString () + "comercios  
        que pierden.");  
        el foreach (Cambian myTrade en el Rendimiento. AllTrades. LosingTrades)  
        {  
            si (myTrade. Entrada. MarketPosition == MarketPosition. Mucho tiempo)  
                Letra (myTrade. ProfitCurrency * myTrade. Cantidad);  
        }  
    }  
}
```

GetTrades () Método

Definición

Devuelve un objeto de TradeCollection que representa todos los comercios que arreglan la posición especificada.

Valor de retorno de método Un objeto de TradeCollection.

Sintaxis <TradeCollection>.GetTrades (instrumento de cuerda, **ensarte** entrySignalName, caso **internacional**)

Parámetros

instrumento Un nombre de instrumento, como "MSFT" entrySignalName El nombre de su entrada señala el caso El suceso para comprobar (1 es la posición más reciente, 2 es el 2do la posición más reciente etc...)

Ejemplos

```
TradeCollection myTrades = Rendimiento.AllTrades ("MSFT", "myEntrySignal", 1);  
la Letra ("La última posición consistió" + myTrades.Conde + "comercios.");
```

p319

www.ninjatrader.com
© NinjaTrader, LLC, 2005

-0-

Clase de TradesPerformance

Perfil de Rendimiento **de definición** de una [colección](#) de objetos [Comerciales](#).

Métodos y

Propiedades

AvgBarsInTrade

AvgTradeDuration

Dinero de comisión [GetSharpeRatio \(\)](#) GrossLoss GrossProfit MaxConsecLoser

MaxConsecWinner

MaxTime2Recover

Por ciento Señala ProfitFactor SharpeRatio

TradesPerDay

Ejemplos

Un [doble](#) valor que representa el número avg de barras por comercio Un valor [de TimeSpan](#) <http://msdn2.microsoft.com/en-us/library/system.timespan.aspx> [representación](#) de la duración [avg cargada por la cantidad](#) de un comercio Un [doble](#) valor que representa la comisión total Consigue un objeto [de TradesPerformanceValues](#) en el dinero

Un [doble](#) valor que representa la pérdida gruesa Un [doble](#) valor que representa el beneficio

bruto Un valor [internacional](#) que representa el número máximo de pérdidas consecutivas

vistas Un valor [internacional](#) que representa el número máximo de

ganadores consecutivos

<<http://msdn2.microsoft.com/en-us/library/system.timespan.aspx>> visto Un valor

de [TimeSpan](#) que representa el tiempo máximo para reponerse de un empate abajo Se pone

un objeto [de TradesPerformanceValues](#) en el por ciento Consigue un objeto [de](#)

[TradesPerformanceValues](#) en puntos Un [doble](#) valor que representa el factor de ganancia Un

[doble](#) valor que representa la proporción de Sharpe usando una vuelta sin riesgo cero Un valor

[internacional](#) que representa los comercios de avg por día

```
protegido anulan OnBarUpdate vacío () {  
    //Sólo comercie si usted tiene menos de 5 perdedores consecutivos en fila  
    si (Performance.RealtimeTrades.TradesPerformance.MaxConsecLoser <5) {  
        //Cambie la lógica aquí  
    }  
}
```

GetSharpeRatio () Método

Definición

Devuelve un objeto de TradeCollection que representa todos los comercios que arreglan la posición especificada.

Valor de retorno de método Un objeto de TradeCollection.

Sintaxis

p320

www.ninjatrader.com

© NinjaTrader, LLC, 2005

<[TradeCollection](#)>.GetTrades (instrumento de cuerda, [ensarte](#) *entrySignalName*, caso [internacional](#))

Parámetros

instrumento Un nombre de instrumento, como "MSFT" entrySignalName El nombre de

su entrada señala el caso El suceso para comprobar (1 es la posición más reciente, 2

es el 2do

la posición más reciente etc...)

Ejemplos

```
TradeCollection myTrades = Rendimiento. AllTrades ("MSFT", "myEntrySignal", 1);  
la Letra ("La última posición consistió" + myTrades. Conde + "comercios.");
```


Clase de TradesPerformanceValues

Valores de Rendimiento **de definición** de una [colección](#) de objetos [Comerciales](#).

- El dinero y el Punto los cálculos basados son por comercio
- el Por ciento cálculos basados son por unidad cambiada

	Métodos y	Propiedades
AvgEtd		Un doble valor la representación avg termina el empate comercial abajo
AvgMae		Un doble valor la representación avg máximo adverso excursión
AvgMfe		Un doble valor la representación avg máximo favorable excursión
AvgProfit		Un doble valor representación avg ganancia
CumProfit		Un doble valor la representación de la ganancia acumulada (el por ciento es compuesto)
DrawDown		Un doble valor representación final de empate comercial abajo
LargestLoser		Un doble valor representación la pérdida más grande
LargestWinner		Un doble valor representación la ganancia más grande
ProfitPerMonth		Un doble valor representación ganancia por mes siempre como a por ciento

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    //Si la ganancia de comercios de tiempo real es> 1000$ dejan de comerciar  
    si (Performance.RealtimeTrades.TradesPerformance.Currency.CumProfit>  
1000)  
        vuelva;  
}
```

-0-

p321

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Métodos de posición y Propiedades

AvgPrice

La definición Consigue el precio medio de una posición de estrategia.

Valor de propiedad Un **doble** valor que representa el precio medio de la posición por unidad.

Posición **de sintaxis**. AvgPrice

-0-

GetProfitLoss ()

La definición Calcula PnL no realizado para la posición.

Valor de retorno de método Un **doble** valor que representa PnL no realizado.

Posición de sintaxis. GetProfitLoss (doblan *marketPrice*, *PerformanceUnit* *performanceUnit*)

Parámetros

marketPrice El precio solía calcular PnL tal como Cerca [0] *performanceUnit*
PerformanceUnit. Dinero
PerformanceUnit. Por ciento
PerformanceUnit. Puntos

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    //Si no el piso imprime nuestro PnL no realizado si (Posición.
```

```

;MarketPosition! = MarketPosition. Piso)
    La letra ("Abren PnL:" + Posición. GetProfitLoss (Cerca [0],
PerformanceUnit. Puntos));}

```

-o-

MarketPosition

La definición Consigue la posición en el mercado corriente.

Valor de propiedad MarketPosition. MarketPosition llano. Long MarketPosition. Corto

Posición **de sintaxis**. MarketPosition

Ejemplos

p322

www.ninjatrader.com
© NinjaTrader, LLC, 2005

```

protegido anulan OnBarUpdate vacío () {
    //Si no el piso imprime nuestro PnL abierto si (Posición. ;MarketPosition! =
    MarketPosition. Piso)
        La letra ("Abren PnL:" + Posición. GetProfitLoss (Cerca [0],
PerformanceUnit. Puntos));}

```

-o-

Cantidad

La definición Consigue la talla de situación actual.

Valor de propiedad Un valor [internacional](#) que representa la talla de posición.

Posición **de sintaxis**. Cantidad

Ejemplos

```

protegido anulan OnBarUpdate vacío () {
    //Imprime la Letra de posición en el mercado corriente (Posición.
    MarketPosition. ToString () + "" +
    Posición. Cantidad. ToString ());}

```

-o-

Posiciones

La definición Sostiene una serie de objetos de Posición que representan posiciones manejadas por la estrategia. Esta propiedad sólo debería ser usada cuando su estrategia ejecuta pedidos contra [instrumentos múltiples](#).

El valor de índice está basado en la serie de objetos de Barras añadidos vía [Añadir \(\)](#)

método. Por ejemplo:

Barras primarias son ES 1 Minuto Barras Secundarias son Terceras Barras De 3 minutos ES es ER2 1 Minuto

Posiciones [0] == Posiciones de posición de ES [1] == Siempre una posición llana, la posición de ES siempre será Posiciones [0] Posiciones [2] == posición de ER2

Valor de propiedad Una serie de objetos de Posición.

Posiciones de sintaxis [índice [internacional](#)]

Propiedad De Estrategia de Encargo

Ejemplos

p323

www.ninjatrader.com
© NinjaTrader, LLC,
2005

```
protegido anulan el vacío Inicializan () {
    Añada (Instrumento, PeriodType. Minuto, 5); Añada ("ER2 06-07, PeriodType.
    Minuto, 5);
}

protegido anulan OnBarUpdate vacío () {
    La letra ("Posición primaria es" + Posiciones [0].MarketPosition); la Letra
    ("posiciones de ER2 es" + Posiciones [2].MarketPosition);

    //Enfoque alternativo. Comprobando lo que Excluye el objeto llama el
    OnBarUpdate ()
    //método, podemos usar sólo la propiedad de Posición desde el que señala al
    correcto
    //posición. si (BarsInProgress == 0)
        La letra ("Posición primaria es" + Posición. MarketPosition);
    más si (BarsInProgress = 2)
        La letra ("posición de Er2 es" + Posición. MarketPosition);
}
```

-0-

AccountSize

La definición Un valor de realización solía calcular la talla de pedido de pedidos de entrada de estrategia. La talla de pedido es dinámicamente calculada basada en esta propiedad sólo poniendo la "Nueva Estrategia" ventana de diálogo durante backtest o en una carta de tiempo real "Propiedad" de cantidad de pedido de juego a "por la talla de cuenta".

- Acciones ordinarias - talla de parte de Pedido es calculada basada en AccountSize / Precio Cercano Corriente
- Futures y Forex - talla de contrato/parte de Pedido es calculada basada en AccountSize / Margen

Requisito (juego de margen por instrumento en el [gerente de Instrumento](#))

Valor de propiedad Un valor [internacional](#) que representa la talla de cuenta.

Sintaxis AccountSize

Ejemplos

```
protegido anulan el vacío Inicializan () {  
    AccountSize = 10000;  
}
```

-0-

Añada ()

La definición Esto es el método proporciona dos funciones:

1. Añade que unas Barras se oponen a una estrategia de desarrollar la estrategia de multiinstrumento o una multimarca de tiempo. Dirigiendo una estrategia, NinjaTrader UI le incitará a seleccionar el instrumento primario e intervalo de barra para correr en. Este objeto de Barras primario llevará un índice [de BarsInProgress](#) de 0. En una multimarca de tiempo y estrategia de multiinstrumento, objetos de Barras suplementarios

p324

www.ninjatrader.com
© NinjaTrader, LLC, 2005

son añadidos vía este método en [Inicializar \(\)](#) método y dados un valor de índice de BarsInProgress incrementado. Ver la información adicional sobre [estrategias de multibarras](#) que corren.

2. Añade un indicador a la estrategia sólo para la demostración de ello en una carta.

*Note: Añadir () método **sólo** debería ser llamado dentro de la estrategia Initialize () el método.*

La sintaxis [Añade \(Indicador de indicador\)](#)

La sintaxis siguiente añadirá otro las Barras se oponen a la estrategia para el instrumento primario de la estrategia. [Añada \(PeriodType periodType, período internacional\)](#)

La sintaxis siguiente permite que usted añada otro objeto de Barras para un instrumento diferente a la estrategia. [Añada \(instrumento de cuerda, PeriodType periodType, período internacional\)](#)

NOTE: Usted puede añadir opcionalmente el nombre de cambio como un sufijo al nombre de símbolo. Esto sólo es aconsejado si el instrumento tiene cambios posibles múltiples en los cuales puede comerciar y es configurado dentro del gerente de Instrumento. Por ejemplo: **Añada ("MSFT Arca", PeriodType. Minuto, 5).**

Parámetros

indicador Un instrumento de objeto de indicador Un nombre de instrumento, como "MSFT" periodType El tipo de período de la barra, como:

PeriodType. Día PeriodType. Minuto PeriodType.

Marca PeriodType. Volumen

período El intervalo de período tal como "3" para barras de 3 minutos

Ejemplos

p325

www.ninjatrader.com

m

© NinjaTrader, LLC,
2005

```
protegido anulan el vacío Inicializan () {  
    //Añada un objeto de Barras de 5 minutos - el índice de BarsInProgress = 1 Añade  
    (PeriodType. Minuto, 5);  
  
    //Añada un 100 objeto de Barras de marca para el ES 12-06 contrato -  
    BarsInProgress  
    índice = 2  
    Añada ("ES 12-06", PeriodType. Marca, 100);  
  
    //Traza un 20 período que el valor medio móvil simple a la carta Añade (SMA (20));  
}  
  
protegido anulan OnBarUpdate vacío () {
```

```

//No haga caso de acontecimientos de actualización de barra para el objeto de
Barras suplementario añadido encima si (BarsInProgress == 1 || BarsInProgress
== 2)
    vuelva;

//Vaya mucho tiempo si tenemos tres barras en todos los objetos de barras si
(Cerca [0]> Abierto [0] && los Finales [1] [0]> Se abren [1] [0] && Finales [2]
[0]>
Se abre [2] [0])
    EnterLong ();
}

```

-0-

BarsArray

BarsArray **de definición** es una serie sosteniendo objetos de Barras que son añadidos a una estrategia vía [Añadir \(\)](#) método. Los objetos de barras pueden ser usados como introducido para [métodos de indicador](#). Esta propiedad es del valor primario trabajando con [estrategias de multiinstrumento o multimarca de tiempo](#).

Valor de propiedad Una serie de objetos de Barras.

Sintaxis BarsArray [*índice internacional*]

Ejemplos

```

protegido anulan el vacío Inicializan () {
    //Añada un objeto de Barras de 5 minutos que es añadido a BarArray//que tomará
    el índice 1 ya que el objeto de Barras primario de la estrategia//será el índice
    0 Añaden (PeriodType. Minuto, 5);
}

```

```

protegido anulan OnBarUpdate vacío () {
    //No haga caso de acontecimientos de actualización de barra para el objeto de
    Barras suplementario añadido encima si (BarsInProgress == 1)
        vuelva;

    /////si
}

```

```

El pase en unas Barras se opone como introducido para los Controles de método
de valor medio móvil simples si 20 SMA de las Barras primarias son mayores que
20 SMA de las Barras secundarias añadidas encima (SMA (20) [0]> SMA (BarsArray
[1], 20) [0])
    EnterLong ();
}

```

p326

www.ninjatrader.co

m

© NinjaTrader, LLC,

2005

-0-

BarsInProgress

Definición

Un valor de índice del objeto de Barras corriente que ha llamado [OnBarUpdate \(\)](#) método. En una estrategia de multibarras, OnBarUpdate () el método es llamado para cada objeto de Barras de una estrategia. Esta flexibilidad permite que usted separe la lógica comercial de acontecimientos de barra diferentes. **En una estrategia Bars sola esta propiedad siempre devolverá un valor de índice de 0 representación de las Barras primarias e instrumentará la estrategia corre en.**

Ver la información adicional sobre [estrategias de multibarras](#) que corren.

El Valor de propiedad Un valor [internacional](#) representa el objeto de Barras que llama OnBarUpdate () método.

Sintaxis BarsInProgress

Ejemplos

```
//Deja suponen que esta estrategia de muestra fuera añadida a un ES 12-06 carta de 1 minuto

protegido anulan el vacío Inicializan () {
    //Añada un objeto de Barras de 5 minutos: el índice de BarsInProgress = 1 Añade
    (PeriodType. Minuto, 5);
}

protegido anulan OnBarUpdate vacío () {
    //Compruebe qué objeto de Barras llama OnBarUpdate () método si (BarsInProgress
    == 0) {
        //Un valor del cero representa las Barras primarias que es el ES
        12-06//carta de 1 minuto.//Hacen algo dentro del contexto de las Barras de
        1 minuto aquí

    } más si (BarsInProgress == 1) {
        //Un valor de 1 representa las barras de 5 minutos secundarias añadidas
        en el
        Initaliaze ()
        //Haga algo dentro del contexto de las Barras de 5 minutos
    }
}
```

-0-

BarsPeriods

La definición Sostiene una serie de objetos de Período sincronizados al número de objetos de Barras únicos sostenidos dentro del objeto de padre NinjaScript. Si una estrategia sostiene dos series de Barras, entonces BarsPeriods sostendrá dos objetos de Período.

Valor de propiedad Una serie de objetos de Períodos.

Sintaxis BarsPeriods [intervalo barSeriesIndex]

Propiedad De Estrategia de Encargo

Ejemplos

```
protegido anulan el vacío Inicializan () {  
    //Añade que unas Barras de 5 minutos se oponen a la estrategia y es  
    automáticamente  
    adjudicado  
    //un índice de objeto de Barras de 1 desde los datos primarios la estrategia  
    es dirigido  
    contra  
    //puesto por el UI toma el índice de 0. Añada (Instrumento, PeriodType. Minuto,  
    5);  
}  
  
protegido anulan OnBarUpdate vacío () {  
    //Copia impresa 5, el valor de las barras secundarias se opone si (BarsInProgress  
    == 1)  
    Letra (BarsPeriods [1].Value);  
}
```

-0-

BarsPeriod

La definición Las Barras primarias objeta el tipo de período y el intervalo.

Valor de propiedad Un objeto de Período.

Sintaxis BarsPeriod. Id BarsPeriod. Valor

* Id puede ser PeriodType de valor. Marca, PeriodType. Volumen, PeriodType. En segundo lugar, PeriodType. Variedad, PeriodType. Minuto, PeriodType. Día, PeriodType. Semana, PeriodType. Mes, PeriodType. Año * Valor devolverá un valor entero

Propiedad De Estrategia de Encargo

Ejemplos

```
//Sólo cuente si hay una 100 carta de marca o mayor protegido anulan OnBarUpdate  
vacío () {  
    si (BarsPeriod. Id == PeriodType. Marca && BarsPeriod. Valor> = 100) {  
        //Lógica de cálculo de indicador aquí  
    }  
}
```

-0-

BarsRequired

Definición

p328

www.ninjatrader.com
© NinjaTrader, LLC, 2005

El número de barras históricas requeridas antes de la estrategia comienza a tratar llamadas a [OnBarUpdate \(\)](#) método. Esta propiedad es generalmente puesta vía el UI comenzando una estrategia.

OnBarUpdate () método no es provocado hasta [CurrentBar](#) > = BarsRequired.

Valor de propiedad Un valor [internacional](#) que representa el número de barras históricas.

Sintaxis BarsRequired

-0-

BarsSinceEntry ()

La definición Devuelve el número de barras que han pasado desde la última entrada especificada.

El Valor de retorno de método Un valor [internacional](#) que representa varias barras.

BarsSinceEntry **de sintaxis** () [BarsSinceEntry \(ensartan signalName\)](#)

La firma de método siguiente debería ser usada trabajando con [estrategias de instrumento y multimarca de tiempo](#):

[BarsSinceEntry \(intervalo barsInProgressIndex, ensarte signalName, intervalo entriesAgo\)](#)

Parámetros

signalName El título de señal de un pedido de entrada especificado en un método de entrada de pedido.

Pase en cuerda vacía "" para señal de falta.

el barsInProgressInd El índice de las Barras objeto el pedido de entrada fue presentado contra. Ver

exceptola propiedad [de BarsInProgress](#). entriesAgo Número de hace unas entradas. Pase en 0 para la entrada más reciente.

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    si (CurrentBar <20)  
        vuelva;  
  
    //Sólo entre si al menos 10 barras han pasado desde nuestra última entrada si  
    (BarsSinceEntry () > 10 && CrossAbove (SMA (10), SMA (20), 1))  
        EnterLong ();  
}
```

BarsSinceExit ()

La definición Devuelve el número de barras que han pasado desde la última salida especificada.

El Valor de retorno de método Un valor [internacional](#) que representa varias barras. Un valor de -1 será devuelto si una salida anterior

p329

www.ninjatrader.com

© NinjaTrader, LLC,

2005

no existe.

BarsSinceExit **de sintaxis** () BarsSinceExit (ensartan
signalName)

La firma de método siguiente debería ser usada trabajando con [estrategias de instrumento y multimarca de tiempo](#):

BarsSinceExit (intervalo barsInProgressIndex, ensarte signalName, intervalo exitsAgo)

Parámetros

signalName El título de señal de un pedido de entrada especificado en un método de entrada de pedido.

Pase en cuerda vacía "" para señal de falta.

el barsInProgressInd El índice de las Barras objeto el pedido de entrada fue presentado contra. Ver

exceptola propiedad [de BarsInProgress](#). exitsAgo Número de hace unas salidas. Pase en 0 para la salida más reciente.

Ejemplos

```
protegido anulan OnBarUpdate vacío () {
    si (CurrentBar <20)
        vuelva;

    //Sólo entre si al menos 10 barras han pasado desde nuestra última salida si
    (BarsSinceExit () > 10 && CrossAbove (SMA (10), SMA (20), 1))
        EnterLong ();
}
```

DataSeriesConfigurable

La definición Indica si la estrategia permitirá la configuración de la serie de datos principal que conducirá la estrategia vía la ventana de diálogo de Estrategia comenzando una estrategia de una carta o de la etiqueta de Estrategias de la ventana Control Center. Esta

propiedad sólo es de valor si una estrategia sólo debería ser dirigida en una serie de datos específica.

La propiedad Valora Este propiedad vueltas **verdaderas** si la serie de datos es configurable; por otra parte, **falso**.

Sintaxis `DataSetConfigurable`

-0-

DefaultQuantity

La definición Una variable de talla de pedido que puede ser puesta programaticalmente u overridden vía la Estrategia que determina la cantidad de un pedido de entrada.

El Valor de propiedad Un valor [internacional](#) representa el número de contratos o partes para entrar en una posición con.

Sintaxis `DefaultQuantity`

p330

www.ninjatrader.com
© NinjaTrader, LLC, 2005

-0-

EntriesPerDirection

Definición

Indica el número máximo de entradas permitidas por dirección mientras una posición está activa basado en la propiedad [de EntryHandling](#).

El Valor de propiedad Un valor [internacional](#) representa el número máximo de entradas permitidas.

Sintaxis `EntriesPerDirection`

Ejemplos

```
//El ejemplo #1//Si un puesto vacante ya existe, EnterLong subsecuente () llamadas  
no es ignorado. protegido anulan el vacío Inicializan () {  
    EntriesPerDirection = 1; EntryHandling = EntryHandling. AllEntries;  
}
```

```
protegido anulan OnBarUpdate vacío () {  
    si (CrossAbove (SMA (10), SMA (20), 1)  
        EnterLong ("Entrada de Cruz de SMA");  
  
    si (CrossAbove (RSI (14, 3), 30, 1)  
        EnterLong ("Entrada de Cruz de RSI");}
```

```
//El ejemplo #2//EnterLong () será tratado una vez para cada entrada únicamente  
llamada. protegido anulan el vacío Inicializan () {
```

```

        EntriesPerDirection = 1; EntryHandling = EntryHandling. UniqueEntries;
    }

    protegido anulan OnBarUpdate vacío () {
        si (CrossAbove (SMA (10), SMA (20), 1)
            EnterLong ("Entrada de Cruz de SMA");

        si (CrossAbove (RSI (14, 3), 30, 1)
            EnterLong ("Entrada de Cruz de RSI");
    }

```

-o-

EntryHandling

La definición Pone la manera en como los pedidos de entrada son manejados. De ser puesto a "EntryHandling. AllEntries", NinjaScript tratará todos [los métodos de entrada de pedido](#) hasta las entradas aceptables máximas puestas por la propiedad [de EntriesPerDirection](#) ha sido alcanzado mientras en un puesto vacante. De ser puesto a "EntryHandling. UniqueEntries", NinjaScript tratará métodos de entrada de pedido hasta las entradas aceptables máximas puestas por la propiedad de EntriesPerDirection por cada uno únicamente llamado

p331

www.ninjatrader.com

entrada.

Valor de propiedad EntryHandling. AllEntries EntryHandling. UniqueEntries

Sintaxis EntryHandling

Ejemplos

```

//Ejemplo #1
© NinjaTrader, LLC, 2005
//Permitirá un máximo de dos entradas mientras una posición está abierta protegido
anulan el vacío Inicializan () {
    EntriesPerDirection = 2; EntryHandling = EntryHandling. AllEntries;
}

protegido anulan OnBarUpdate vacío () {
    si (CrossAbove (SMA (10), SMA (20), 1)
        EnterLong ("Entrada de Cruz de SMA");}

//El ejemplo #2//EnterLong () será tratado una vez para cada entrada únicamente
llamada. protegido anulan el vacío Inicializan () {
    EntriesPerDirection = 1; EntryHandling = EntryHandling. UniqueEntries;
}

protegido anulan OnBarUpdate vacío () {
    si (CrossAbove (SMA (10), SMA (20), 1)
        EnterLong ("Entrada de Cruz de SMA");

```

```

    si (CrossAbove (RSI (14, 3), 30, 1)
        EnterLong ("Entrada de Cruz de RSI");
}

```

-o-

ExitOnClose

La definición Indica si la estrategia anulará los pedidos generados de toda la estrategia y cerrar todas las posiciones de estrategia abiertas en el final de la sesión. Esta propiedad puede ser puesta programatically en [Inicializar \(\)](#) método o conducida por el UI en el tiempo de ejecución. También ver ["ExitOnCloseSeconds"](#).

La propiedad Valora Este propiedad vueltas **verdaderas** si la estrategia saldrá en el final; por otra parte, **falso**.

Sintaxis ExitOnClose

Ejemplos

p332

www.ninjatrader.com
© NinjaTrader, LLC,
2005

```

protegido anulan el vacío Inicializan () {
    //Provoca la salida en la función cercana 30 segundos antes del final de
    sesión
    ExitOnClose      =
    verdadero;
    ExitOnCloseSeconds = 30;}

```

-

o-

ExitOnCloseSeconds

La definición El número de segundos antes del fin del tiempo de sesión actual que la función ["de ExitOnClose"](#) provocará. Esta propiedad puede ser puesta programatically en [Inicializar \(\)](#) método o conducida por el UI en el tiempo de ejecución.

Valor de propiedad Un valor [internacional](#) que representa el número de segundos.

Sintaxis ExitOnCloseSeconds

Ejemplos

```

protegido anulan el vacío Inicializan () {
    //Provoca la salida en la función cercana 30 segundos antes del final de sesión
    ExitOnClose      =

```

```
verdadero;  
ExitOnCloseSeconds = 30;}
```

O-

IncludeCommission

La definición Indica si los resultados de rendimiento de estrategia incluirán la comisión de backtest histórico.

La propiedad Valora Este propiedad vueltas **verdaderas** si la estrategia incluye la comisión de backtest histórico; por otra parte, **falso**.

Sintaxis IncludeCommission

Ejemplos

```
protegido anulan el vacío Inicializan () {  
    IncludeCommission = verdadero;  
}
```

-O-

RealtimeErrorHandling

La definición Define el comportamiento de una estrategia cuando el pedido generado de una estrategia es devuelto del

p333

www.ninjatrader.com
© NinjaTrader, LLC, 2005

el servidor del agente de bolsa en un estado "Rechazado".

- El ajuste de este valor de propiedad a "TakeNoAction" puede tener ***serious* adverso** afecta en a
la marcha de la estrategia a menos que usted haya programado su propio rechazo de pedido que se maneja en OnOrderUpdate () método
- El manejo de rechazo definido del usuario es avanzado y si ***only*** ser dirigido por programadores **con experiencia**

Valor de propiedad RealtimeErrorHandling. TakeNoAction
RealtimeErrorHandling.StopStrategyCancelOrdersClosePositions (Valor por defecto de una estrategia)

Sintaxis RealtimeErrorHandling

Ejemplos

```
protegido anulan el vacío Inicializan () {  
    RealtimeErrorHandling = RealtimeErrorHandling. TakeNoAction;}
```

```

IOrder privado stopLossOrder = nulo;

protegido anulan OnBarUpdate vacío () {
    si (entryOrder == nulo && Cerca [0]> Abierto [0])
        EnterLong ();

    si (stopLossOrder == nulo)
        stopLossOrder = ExitLongStop (Posición. AvgPrice - 10 * TickSize);}

protegido anulan OnOrderUpdate vacío (pedido de IOrder) {
    si (!stopLossOrder && stopLossOrder.Señal == pedido.Señal) {
        //Manejo de rechazo si (pedido.OrderState == OrderState.Rechazado) {
            //¡Párese el pedido de pérdida fue rechazado!!!!//Hacen algo sobre ello
            aquí
        }
    }
}

```

-0-

Resbalamiento

Los Juegos **de definición** la cantidad de resbalamiento en marcas usados en cálculos de rendimiento durante backtests.

Valor de propiedad Un valor **internacional** que representa las marcas de número.

Resbalamiento **de sintaxis**

p334

www.ninjatrader.com
© NinjaTrader, LLC,

2005

Ejemplos

```

protegido anulan el vacío
Inician () {
    Resbalamiento = 2;}

```

-

0-

TimeInForce

Definición

Pone el tiempo propiedad vigente para todos los pedidos generados por una estrategia. Esta propiedad es la persona a cargo en a que hora vigente su agente de bolsa puede o puede no apoyar.

Valor de propiedad Cbi. TimeInForce. Día Cbi. TimeInForce. Gtc

Sintaxis TimeInForce

Ejemplos

```
protegido anulan el vacío Inicializan () {  
    TimeInForce = Cbi. TimeInForce. Día;  
}
```

-0-

TraceOrders

Definición Cuando permitido, los rastros son generados y mostrados en la ventana NinjaScript Output para cada llamada de una confirmación de suministro [de método de pedido](#) que el método es entrado e información que provee si los métodos de pedido no son ignorados y por qué. Esto es valioso para eliminar fallos si usted no ve el comportamiento esperado llamando un método de pedido. Esta propiedad puede ser puesta programatically en [Inicializar \(\)](#) método.

La salida se referirá a un método "PlaceOrder ()" que es un método interno en el cual todos Entran () y Salida () uso de métodos.

La propiedad Valora Este propiedad vueltas **verdaderas** si la estrategia va la información de rastro de salida; por otra parte, **falso**.

Sintaxis TraceOrders

Ejemplos

```
protegido anulan el vacío Inicializan () {  
    TraceOrders      = verdadero;  
}
```

-0-

p335

www.ninjatrader.com
© NinjaTrader, LLC, 2005

WaitForOcoClosingBracket

La definición Allí puede ser agentes de bolsa que requieren que los pedidos de OCO sean presentados simultáneamente en una llamada de API sola contra el envío de ellos en la secuencia con el identificador OCO definido de un usuario incluir. Para agentes de bolsa que requieren que pedidos de OCO sean presentados en una llamada a la función sola, una estrategia NinjaScript debe esperar hasta que ella tenga ambas piernas del par OCO generado por [SetStopLoss \(\)](#), [SetTrailStop \(\)](#) y [SetProfitTarget \(\)](#).

Si usted sólo desea enviar un objetivo de ganancia o pérdida de parada (pero no ambos) vía alguno del Juego... () métodos mencionados anteriormente, cuando permiten WaitForOcoClosingBracket, sus pedidos de salida nunca serán enviados ya que NT tiene que

esperar hasta que esto tenga ambos pedidos del soporte OCO. Si usted deja inválido `WaitForOcoClosingBracket`, NinjaTrader presentará inmediatamente una parada o sacará ganancia el pedido objetivo, cualquiera es presentado primero.

Agentes de bolsa afectados corrientes:

TD AMERITRADE

* Para cualquier otro agente de bolsa, esta propiedad no tiene afectan.

La propiedad Valora Este propiedad vueltas **verdaderas** si la estrategia esperará ambas piernas de un soporte OCO a ser llamada en una estrategia antes de presentar al par de pedido al agente de bolsa; por otra parte, **falso**.

Sintaxis `WaitForOcoClosingBracket`

Ejemplos

```
protegido anulan el vacío Inicializan () {  
    WaitForClosingOcoBracket = falso;}
```

-o-

Métodos de Indicador de encargo y Propiedades

Descripción

Los métodos y las propiedades cubiertas en esta sección son únicos para el desarrollo de indicador de encargo. Las propiedades de configuración de indicador globalmente definen varios comportamientos de indicadores. Todas las propiedades tienen valores por defecto y pueden ser anuladas poniéndolos en `Inicializar ()` el método del indicador.

-o-

Añada ()

La definición Añade complot y objetos de línea que definen como una serie de datos de indicador es visualizada en una carta. Cuando este método es llamado para añadir un complot, un objeto [de DataSeries](#) asociado es creado que es sostenido en la colección [de Valores](#).

Sintaxis

p336

www.ninjatrader.com
© NinjaTrader, LLC, 2005

[Añada \(Complot de complot\)](#) [Añaden \(Línea de línea\)](#)

* *Añadir ()* el método **sólo** debería ser llamado dentro del indicador *Initialize ()* el método.

Parámetros

complot	Una línea de objeto de
Complot	Un objeto de Línea

Ejemplos

```
//Añade que un complot de estilo de línea azul Añade
(nuevo Complot (Color. Azul, "myPlot"));

//Añade que un complot de estilo de historgram azul
Añade (nuevo Complot (Color. Azul, PlotStyle. Barra,
"myPlot"));

//Añade que un complot de estilo de línea azul con un
grosor de 3 Añade (nuevo Complot (nueva Pen (Color.
Azul, 3), "myPlot"));

//Añade que una línea de oscilador en un valor de 30
Añade (nueva Línea (Color. Gris, 30, "más Abajo"));
```

Puntas 1. Hay muchas variaciones de complot definido y objetos de línea que no son documentados. Aconsejamos usar al mago de indicador NinjaScript para generar su complot de indicador y líneas de oscilador.

2. Los objetos de complot no sostienen los valores de indicador actuales. Ellos simplemente definen como los valores del indicador son trazados en una carta.

3. Un indicador puede calcular valores múltiples y por lo tanto sostener complots múltiples de determinar la demostración de cada valor deliberado. Los valores de indicador son sostenidos en la colección [de Valores](#) del indicador.

4. Si su indicador calcula y almacena tres valores entonces conspira son adjudicados según el ejemplo abajo.

```
Protegido anulan el vacío Inicializan () {
    //Añada tres complots y se asoció los objetos de DataSeries Añaden (nuevo
    Complot (Color. Azul, "PlotA")); //Define el complot para Valores [0]
    Añada (nuevo Complot (Color. Rojo, "PlotB")); //Define el complot para
    Valores [1]
    Añada (nuevo Complot (Color. Verde, "PlotC")); //Define el complot para
    Valores [2]}
```

-0-

AutoScale

Definición De ser verdadero, el complot (s) de indicador es incluido para determinar el

escalamiento de y de la carta.

La propiedad Valora Este propiedad vueltas **verdaderas** si el complot (s) de indicador es incluido en la escala de y; por otra parte, **falso**. Juego de falta a **verdadero**.

p337

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Sintaxis AutoScale

Ejemplos

```
//Inicialice el método de un indicador de encargo protegido anulan el vacío
Inician () {
    Añada (nuevo Complot (Color. Naranja, "SMA")); AutoScale = falso;
}
```

-0-

BackColorAll

La definición Elige el color trasero para todos los paneles de carta para la barra corriente.

La propiedad Valora una estructura [en color](#) que representa el color de esta barra de precios.

Sintaxis BackColorAll

Ejemplos

```
//Elige el color trasero a BackColorAll verde pálido = Color. PaleGreen;

//Elige el color trasero al juego de color de falta en la ventana BackColorAll de
diálogo de propiedades de carta = Color. Vacío;

//¿Elige el color trasero al rosado cuándo la cotización de cierre está menos que
el 20 período SMA//y al verde lima cuando encima (ver la imagen abajo) BackColorAll
= SMA (20) [0]> = Cerca [0]? Color. Rosado: Color. PaleGreen;
```

-0-

BarsRequired

Definición

El número de barras en una carta requerida antes de los complots de indicador. Esto no es el mismo como un número mínimo de barras requeridas calcular los valores de indicadores.

La propiedad Valora un valor [internacional](#) que representa el número mínimo de barras requeridas.

Sintaxis BarsRequired

Ejemplos

```
//Inicialice el método de un indicador de encargo protegido anulan el vacío  
Inician () {  
    Añada (nuevo Complot (Color.Naranja, "SMA")); BarsRequired = 10; //no conspiran  
    hasta la 11ra barra en la carta
```

}

-o-

DayBreakLines

El día de Complots **de definición** rompe líneas en el panel de indicador.

La propiedad Valora Este propiedad vueltas **verdaderas** si las líneas de ruptura de día son trazadas en el panel de indicador; por otra parte, **falso**. Juego de falta a **verdadero**.

Sintaxis DayBreakLines

p339

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Ejemplos

```
//Inicialice el método de un indicador de encargo protegido anulan el vacío
Inician () {
    Añada (nuevo Complot (Color. Naranja, "SMA")); DayBreakLines = falso;//las
    líneas de ruptura de Día no conspirarán en el indicador
panel}
```

Puntas 1. La carta paternal del panel de indicador tiene una propiedad similar 'DayBreakLines' que de ser puesto al falso, anulará el ajuste local del indicador de **ser verdadero**.

-o-

Desplazamiento

La definición Un valor de compensación que cambia el valor trazado de un indicador.

La propiedad Valora un valor **internacional** que representa el número de hace unas barras

Desplazamiento **de sintaxis**

Ejemplos

```
//Inicialice el método de un indicador de encargo protegido anulan el vacío
Inician () {
    Añada (nuevo Complot (Color. Naranja, "SMA")); Desplazamiento = 2;//Complots
    el valor de indicador de hace 2 barras en el
barra corriente}
```

-o-

DisplayInDataBox

Definición De ser verdadero, los valores de complot (s) de indicador son mostrados en la

caja de datos de carta.

La propiedad Valora Este propiedad vueltas **verdaderas** si los valores de complot (s) de indicador son mostrados en la caja de datos de carta; por otra parte, **falso**. Juego de falta a **verdadero**.

Sintaxis DisplayInDataBox

Ejemplos

```
//Inicialice el método de un indicador de encargo protegido anulan el vacío
Inician () {
    Añada (nuevo Complot (Color. Naranja, "SMA")); DisplayInDataBox = falso;
}
```

p340

www.ninjatrader.com
© NinjaTrader, LLC, 2005

-0-

DrawOnPricePanel

Definición

De ser verdadero, todos dibujan objetos son pintados en el panel de precios que es el panel que muestra los datos de barra de precios. De ser falso, dibuje objetos son pintados en el panel de indicador actual sí mismo.

La propiedad Valora Este propiedad vueltas **verdaderas** si las pinturas de indicador dibujan objetos en el panel de precios; por otra parte, **falso**. Juego de falta a **verdadero**.

Sintaxis DrawOnPricePanel

Ejemplos

```
//Inicialice el método de un indicador de encargo protegido anulan el vacío
Inician () {
    Añada (nuevo Complot (Color. Naranja, "SMA")); DrawOnPricePanel =
    falso;//Dibujan objetos ahora pintan en el panel de indicador
    sí mismo}
}
```

-0-

HorizontalGridLines

Complots **de definición** líneas de rejilla horizontales en el panel de indicador.

La propiedad Valora Este propiedad vueltas **verdaderas** si las líneas de rejilla horizontales son trazadas en el panel de indicador; por otra parte, **falso**. Juego de falta a **verdadero**.

Sintaxis HorizontalGridLines

Ejemplos

```
//Inicialice el método de un indicador de encargo protegido anulan el vacío
Inician () {
    Añada (nuevo Complot (Color. Naranja, "SMA")); HorizontalGridLines = falso; //no
    conspirará en el panel de indicador
}
```

Puntas

1. La carta paterna del panel de indicador tiene una propiedad similar 'HorizontalGridLines' que de ser puesto al falso, anulará el ajuste local del indicador de **ser verdadero**.

-o-

Clase de línea

Los Objetos **de definición** derivados de la clase de Línea son usados para caracterizar como una línea de oscilador es visualmente mostrada (trazada) en una carta.

Sintaxis

p341

www.ninjatrader.com
© NinjaTrader, LLC,
2005

Línea (Color en color, doble el valor, el nombre de cuerda) la Línea (Pluma de pluma, doble valor, nombre de cuerda)

Parámetros

<<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlfsystemdrawingcolormemberstopic.as>

p>color El color de línea (referencia) valor El uso de valor para definir donde la línea es la pluma trazada La pluma solía dibujar el complot (referencia)

Ejemplos

Ver [Añadir \(\)](#) método para ejemplos.

o-

Líneas

Definición

Una colección sosteniendo todos los objetos de Línea que definen la visualización características las líneas de oscilador del indicador.

Valor de propiedad Una colección de objetos de Línea.

Líneas de sintaxis [[índice internacional](#)]

Ejemplos

```
//Inicialice el método de un indicador de encargo protegido anulan el vacío
Inicializan () {
    //Las líneas son añadidas a la colección de Líneas en el pedido
    Añada (nueva Línea (Color. Gris, 30, "más Abajo"));
    //Almacenado en Líneas [0] Añaden (nueva Línea (Color. Gris, 70,
    "Superior")); //Almacenado en Líneas [1]
}

//Dinámicamente cambie el color de la línea superior y grosor basado en el valor
de indicador protegido anulan OnBarUpdate vacío () {
    si (Valores [0]> 70)
        Líneas [1].Pen = nueva Pen (Color. Azul, 3);
    más
        Complots [1].Pen = nueva Pen (Color. Gris, 1);
}
```

-o-

LinesConfigurable

Definición De ser verdadero, cualquier línea () de indicador es configurable dentro de la ventana de diálogo de indicador

La propiedad Valora Este propiedad vueltas **verdaderas** si cualquier línea () de indicador es configurable; por otra parte, **falso**. Juego de falta a **verdadero**.

Sintaxis LinesConfigurable

Ejemplos

p342

www.ninjatrader.com
© NinjaTrader, LLC, 2005

```
//Inicialice el método de un indicador de encargo protegido anulan el vacío
Inicializan () {
    Añada (nuevo Complot (Color. Naranja, "SMA")); LinesConfigurable = falso; //las
    líneas de Indicador no son configurables
}
```

-o-

Revestimiento

Definición De ser verdadero, el complot (s) de indicador es usado el panel de carta sobre la cumbre del precio.

La propiedad Valora Este propiedad vueltas **verdaderas** si el complot (s) de indicador es usado el panel de carta; por otra parte, **falso**. Juego de falta a **falso**.

Revestimiento **de sintaxis**

Ejemplos

```
//Inicialice el método de un indicador de encargo protegido anulan el vacío
Inician () {
    Añada (nuevo Complot (Color. Naranja, "SMA")); Revestimiento = verdadero; //los
    complots de Indicador son usados el panel de carta encima de
    precio}
```

-0-

PaintPriceMarkers

Definición De ser verdadero, los valores de complot de indicador son mostrados por marcadores de precios en el eje Y.

La propiedad Valora Este propiedad vueltas **verdaderas** si los valores de complot de indicador son mostrados en el eje Y; por otra parte, **falso**. Juego de falta a **verdadero**.

Sintaxis PaintPriceMarkers

Ejemplos

```
//Inicialice el método de un indicador de encargo protegido anulan el vacío
Inician () {
    Añada (nuevo Complot (Color. Naranja, "SMA")); PaintPriceMarkers =
    verdadero; //los valores de complot de Indicador son mostrados en el y
    eje}
```

-0-

Clase de complot

Definición

p343

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Los objetos derrived de la clase de Complot son usados para caracterizar como los valores de indicador son visualmente mostrados (trazados) en una carta.

Complot de sintaxis (*Color en color, ensarte el nombre*) el **Complot** (*Pluma de pluma, nombre de cuerda*) **Complot** (*Color en color, PlotStyle plotStyle, nombre de cuerda*) **Complot** (*Pluma de pluma, PlotStyle plotStyle, nombre de cuerda*)

Complot de propiedades. Minuto - Juegos el valor mínimo requerido para el complot de mostrar Complot. Max - los Juegos el máximo valora el complot mostrarán

* Ver [el "Color de Complot de encargo" tutelar \[vía Umbrales\]\(#\) por ejemplo del Complot.](#) Minuto y Complot. Propiedades de Max.

Parámetros

color

El

trace el color ([referencia](#))

nombre

El

nombre de complot

pluma

El

la pluma solía dibujar el complot ([referencia](#))

plotStyle

El

PlotStyle del complot:

PlotStyle. Barra
PlotStyle. Dot
PlotStyle. Picadillo
PlotStyle. Línea
PlotStyle.
Cuadrado

Los ejemplos Ven [Añadir \(\)](#) método para ejemplos.

o-

Complots

Definición

Una colección sosteniendo todos los objetos de Complot que definen la visualización characteristics del indicador.

Valor de propiedad Una colección de objetos [de Complot](#).

[Complots de sintaxis](#) [[índice internacional](#)]

Ejemplos

```
//Inicialice el método de un indicador de encargo protegido anulan
el vacío Inicializan () {
    //Los complots son añadidos a la colección de Complots en el
    pedido Añaden (nuevo Complot (Color. Naranja,
    "Plot1")); //Almacenado en Complots [0]
    Añada (nuevo Complot (Color. Azul, "Plot2")); //Almacenado
    en Complots [1]}

//Dinámicamente cambie el color del complot primario basado en el valor de indicador
protegido anulan OnBarUpdate vacío () {
    si (Valor [0]> 100)
        Complots [0].Pen = nueva Pen (Color. Azul);
    más
        Complots [0].Pen = nueva Pen (Color. Rojo);
}
```

-o-

PlotsConfigurable

Definición De ser verdadero, cualquier complot () de indicador es configurable dentro de la ventana de diálogo de indicador

La propiedad Valora Este propiedad vueltas **verdaderas** si cualquier complot () de indicador es configurable; por otra parte, **falso**. Juego de falta a **verdadero**.

Sintaxis PlotsConfigurable

Ejemplos

```
//Inicialice el método de un indicador de encargo protegido anulan el vacío
Inicializan () {
    Añada (nuevo Complot (Color. Naranja, "SMA")); PlotsConfigurable = falso; //los
    Complots no son configurables en el indicador
    diálogo}
```

-o-

PriceType

Tipo **de definición** de datos de precios tal como abierto, alto, bajo etc...

El Valor de propiedad Esta propiedad devuelve un valor de PriceType de tampoco:

PriceType. PriceType cercano. PriceType alto. PriceType bajo. PriceType mediano. PriceType abierto. Típico

Sintaxis PriceType

p345

www.ninjatrader.com
© NinjaTrader, LLC,

Ejemplos

```
//Inicialice el método de un indicador de encargo protegido anulan el vacío
Inicializan () {
    Añada (nuevo Complot (Color. Naranja, "SMA")); PriceType = PriceType.
    Alto; //Indicador usará altos precios por cálculos}
```

-o-

Actualización ()

La definición que Este método obligará OnBarUpdate () método a ser llamado de modo que los valores de indicador sean actualizados. Este método sólo es relevante en casos de uso específicos y probablemente sólo usado por programadores avanzados. Cuando los indicadores son introducidos (llamado) dentro de una estrategia NinjaScript, ellos son optimizados sólo para contar cuando los visitan en backtest histórico. Ya que el modelo de indicador NinjaTrader es muy flexible, es posible crear propiedades públicas en un indicador de encargo que los valores de retorno del usuario interno definieron variables. Si estas propiedades requieren que OnBarUpdate () el método sea llamado antes de devolver un valor, incluya una llamada en esta Actualización () método en el comprador de propiedad.

Actualización **de sintaxis** ()

Ejemplos

```
double tripleValue privado = 0;

protegido anulan OnBarUpdate vacío () {
    si (CurrentBar <20)
        vuelva;

    tripleValue = SMA (20) [0] * 3; Valores [0].Set (SMA (20) [0]);
}

el público dobla TripleValue {
    póngase {
        Actualización (); devuelva tripleValue;
    }
}
```

-o-

Valor

El Valor **de definición** se refiere a los 1ros Valores de objeto de DataSeries [0] en el indicador. Esto es el valor de indicador primario.

Valor de propiedad Un objeto [de DataSeries](#).

Valor **de sintaxis**

Ejemplos

```
//El método de OnBarUpdate de un indicador de encargo protegido anula OnBarUpdate  
vacío () {  
    //Comprueba el indicador valor primario 1 barra agao y pone el valor del  
    indicador  
    //para la barra corriente evaluada si (Valor [1] <Alto [0] - Bajo [0])  
    Valor. Juego (Alto [0] - Bajo [0]);  
    más  
    Valor. Juego (Alto [0] - Cerca [0]);}
```

Puntas 1. El objeto de DataSeries referido por esta propiedad es creado llamando [Añadir \(\)](#) método.

-o-

Valores

Los Valores **de definición** son una posesión de colección objetos [de DataSeries](#) que sostienen los valores deliberados subyacentes del indicador. Los objetos de DataSeries sostenidos por esta colección son añadidos llamando [Añadir \(\)](#) método.

Valor de propiedad Una colección de objetos de DataSeries.

Valores **de sintaxis** [[índice internacional](#)]

-o-

VerticalGridLines

Complots **de definición** líneas de rejilla verticales en el panel de indicador.

La propiedad Valora Este propiedad vueltas **verdaderas** si las líneas de rejilla verticales son trazadas en el panel de indicador; por otra parte, **falso**. Juego de falta a **verdadero**.

Sintaxis VerticalGridLines

Ejemplos

```
//Las líneas de rejilla verticales no trazarán en el panel de indicador  
VerticalGridLines = falso;
```

Puntas 1. La carta paternal del panel de indicador tiene una propiedad similar 'VerticalGridLines' que de ser puesto al falso, anulará el ajuste local del indicador de **ser verdadero**.

-0-

Costumbre Se Llena y Tipos de Optimizer

Llene Tipos

La costumbre Se Llena los Algoritmos Además del dos sistema llenan tipos (Falta y Liberal) usted también puede la escritura una costumbre llena algoritmos usando cualquier código/editor de textos.

Copie el archivo "Mis Documentos \<Carpeta de NinjaTrader> \bin\Custom\Type \LiberalFillType.cs" y proporcione un nombre nuevo exluding "el" signo, como "Mis Documentos \<Carpeta de NinjaTrader> \bin\Custom\Type\MyFillType.cs".

Entonces dentro de su redactor de código:

1. Provea un nuevo nombre de demostración por ejemplo renombran "Liberal" a "MyFill" 2. Provea un nuevo nombre de clase por ejemplo renombran "LiberalFillType" a "MyFillType" 3. Cifre su llenar el algoritmo dentro de Llenar () método 4. Una vez completo, abierto cualquier indicador o estrategia dentro del Redactor de NinjaScript de NinjaTrader y compilan. Esto compilará el nuevo llenan el algoritmo y lo ponen a disposición dentro de cualquier ventana de diálogo backtest.

-0-

Tipo de Optimizer

Esta sección está bajo la construcción sin embargo, si usted es un avanzado C# programador usted puede ver el archivo Mis Documentos \<Carpeta de NinjaTrader> \bin\Custom\Type \DefaultOptimizationMethod.cs.

-0-

Métodos de Ajuste y Alarma compartidos

Alarma ()

La definición Genera una alarma visual/audible para la ventana [Alerts](#).

- Las llamadas a este método no son ignoradas en datos históricos, sólo trabajará en el de tiempo real.

p348

www.ninjatrader.com
© NinjaTrader, LLC,

2005

Sintaxis

La alarma (*ensartan id*, *prioridad de Prioridad*, *ensartan el mensaje*, *ensartan soundLocation*, *intervalo rearmSeconds*, *Color backColor*, *Color foreColor*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

id Id único para esta prioridad despierta Cualquiera de los valores siguientes:

NinjaTrader. Cbi. Prioridad. NinjaTrader alto. Cbi. Prioridad. NinjaTrader bajo. Cbi. Prioridad. Medio

mensaje Mensaje despierto soundLocation Nombre de fichero de archivo.wav
para jugar. Proporcione el camino de archivo absoluto o sólo
el nombre si el archivo está localizado en C:\Program Files\NinjaTrader
Instalación carpeta de Folder\sounds

rearmSeconds Establece el número de segundos que una alarma rearmará. Si la misma
alarma

(identificado por el parámetro id) es llamado dentro de una ventana de
tiempo del tiempo de la última alarma + rearmSeconds, el
<<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfssystemdrawingcolormemberstopic.asp>>
la alarma no será ignorada.

backColor La fila despierta atrás colorea ([referencia](#)) foreColor La fila despierta color
delantero ([referencia](#))

Ejemplos

```
//Genera una Alarma despierta ("myAlert", NinjaTrader. Cbi. Prioridad.  
Alto, "Umbral alcanzado", "Alert1.wav", 10, Color. Negro, Color.  
Amarillo);
```

-o-

ClearOutputWindow ()

La definición Limpia todos los datos de la ventana de salida de NinjaTrader

Sintaxis ClearOutputWindow ()

Método De Indicador de Encargo, Estrategia de Encargo

-o-

Tronco ()

La definición Genera la salida de tronco a la [etiqueta de Tronco](#) de ventana NinjaTrader Control Center.

Sintaxis

p349

www.ninjatrader.com
© NinjaTrader, LLC, 2005

El tronco (*ensartan el mensaje*, LogLevel
logLevel)

Método De Indicador de Encargo,
Estrategia de Encargo

mensaje LogLevel

Ejemplos

El Usuario **de parámetros** definió el mensaje de tronco Juegos el nivel de mensaje para el acontecimiento de tronco. Los niveles diferentes son coloreados cifrados en el tronco de NinjaTrader.

NinjaTrader. Cbi. LogLevel. La alarma (también generará una música pop ventana de mensaje con el texto de tronco) NinjaTrader. Cbi. LogLevel. Error NinjaTrader. Cbi. LogLevel. Información NinjaTrader. Cbi. LogLevel. Advertencia

```
//Genera un mensaje de tronco el Tronco ("Esto es un mensaje de tronco",
```

```
NinjaTrader. Cbi. LogLevel. Información);
```

```
//Genera un mensaje de tronco con una ventana Log de notificación ("Esta  
voluntad energate una ventana de notificación también", NinjaTrader. Cbi.  
LogLevel. Alarma);
```

-o-

PlaySound ()

Juegos **de definición** un archivo.wav.

- Las llamadas a este método no son ignoradas en datos históricos, sólo trabajará en el de tiempo real.

PlaySound de sintaxis (ensartan el nombre del archivo)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

nombre del archivo Nombre de fichero de
archivo.wav para jugar

Ejemplos

```
//Juega el archivo wav mySound.wav PlaySound  
("C:\mySound.wav");
```

-o-

p350

www.ninjatrader.com
© NinjaTrader, LLC,

2005

Letra ()

Definición

Datos de letras a la ventana de salida de
NinjaTrader

La **Letra de sintaxis** (ensartan el
mensaje) la **Letra** (bool valor) **Letra** (doble
valor) **Letra** (valor internacional) **Letra**
(valor de DateTime)

Método De Indicador de Encargo,
Estrategia de Encargo

de parámetros Mensaje
El usuario definió el mensaje

Ejemplos

```
//Genera un mensaje la Letra ("Esto es un mensaje");  
  
//Imprime la barra corriente que SMA valoran a la ventana  
Print de salida (SMA (Cerca, 20) [0]);
```

Puntas 1. Usted puede alinear precios para ser formateados para la depuración más fácil tal como Bajo [0].ToString ("0.00"). Tan si usted tiene

12.5 12.75

usted se pondría

12.50 12.75

-o-

ResetAlert ()

Reinicios **de definición** el tiempo rearmar de una alarma creada vía la [Alarma \(\)](#) método.

[ResetAlert](#) **de sintaxis** ([ensartan](#) *id*)

Método De Indicador de Encargo, Estrategia de Encargo

p351

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Parámetros id

El valor de id único que identifica la alarma

-o-

ResetAlerts ()

La definición Reinicializa el tiempo rearmar de todas las alarmas provocadas a través de NinjaTrader sin tener en cuenta donde las alarmas fueron creadas.

Sintaxis ResetAlerts ()

Método De Indicador de Encargo, Estrategia de Encargo

-o-

SendMail ()

La definición Envía un mensaje de correo electrónico.

- Las llamadas a este método no son ignoradas en datos históricos, sólo trabajará en el de tiempo real.

SendMail de sintaxis (ensartan *de*, *cuerda a*, *sujeto de cuerda*, *texto de cuerda*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

de sujetar texto

Ejemplos

De dirección de correo electrónico A Línea de
asunto de dirección de correo electrónico de
cuerpo de mensaje de correo electrónico de
correo electrónico

```
//Genera un mensaje de correo electrónico SendMail ("support@ninjatrader.com",  
"customer@winners.com", "la Alarma Comercial", "Compran ES");
```

Puntas 1. Si los correos no son enviados, compruebe la etiqueta de Tronco de Centro de Control cualquier información de error. Usted también puede tratar de cambiar el servidor de correo sociable SMTP para usar el servidor de correo de su propio ISP. Usted puede poner esta información vía los Instrumentos de ventana Control Center> Opciones> Misc.

-o-

p352

www.ninjatrader.com
© NinjaTrader, LLC,
2005

Métodos de Datos compartidos y Propiedades

Descripción

Los métodos de documentos de siguiente sección y propiedades que acceso varias formas de datos incluso datos de barra, datos de precios y formas estadísticas de datos.

-o-

El funcionamiento con Datos de Precios

El objetivo principal de desarrollar Indicadores de encargo y Estrategias con NinjaScript es evaluar datos de precios. NinjaScript permite que usted se refiera a datos de precios corrientes e históricos. Hay varias categorías de datos de precios que incluyen Tipos de Precios, Indicador y Serie Histórica de Encargo.

Tipo de Precios **de definiciones** - barra Estándar tipos de precios basados, como cierre, apertura, alto, precios bajos e Indicador de volumen - valores Deliberados basados en valores de tipo de precios, como una Costumbre de valor medio móvil simple [DataSeries](#) Histórico - la Costumbre calculó valores que usted desea almacenar y asociar a cada barra histórica

Referirse a Datos de Tipo de Precios

Tipo de precios	Sintaxis	El redactor Shortcut
Definition		
Cerca	Cerca [intervalo "c" + tabulador Último precio cambiado de una barra barsAgo]	
Abierto	Abierto [intervalo "o" + tabulador Precio inicial de una barra barsAgo]	
Alto	Alto [intervalo "h" + tabulador El precio cambiado más alto de una barra barsAgo]	
Bajo	Bajo [intervalo	

"i" + tabulador
El precio cambiado más bajo de una barra

barsAgo]

Volumen

Volumen [intervalo

"v" + tabulador
El número de partes/contratos comerciados de

barsAgo]

una barra

Entrada

Entrada [intervalo

"i" + tabulador
Tipo de precios de *Default de una barra

barsAgo]

Usted notará que para referirse a cualesquiera datos de precios usted tiene que incluir un valor para [intervalo barsAgo]. Esto es un concepto muy simple; el barsAgo representa el número de barras a la referencia y el intervalo indica que barsAgo es un valor entero. Como un ejemplo, podríamos escribir una declaración al control si el alto precio de hace 1 barra es menos que el alto precio de la barra corriente como esto:

Alto [1] < Alto [0];

Usted podría escribir una declaración para calcular la cotización de cierre media de las tres últimas barras como esto:

(Cerca [2] + Cerca [1] + Cerca [0]) / 3;

Como usted puede haber entendido ya, refiriéndose a los datos de barra corrientes es llevado a cabo pasando en un valor de 0 (cero) al parámetro barsAgo. Básicamente, decimos me muestran el

p353

www.ninjatrader.com
© NinjaTrader, LLC, 2005

los datos de precios de hace barras unas cero, el que significa la barra corriente.

** la Entrada es una propiedad especial que puede ser tenida acceso y siempre señala al tipo de precios de falta. Si usted tiene un indicador que apoya el tipo de precios diferente que usa como su fuente de datos, si usted se refiere a la Entrada [en barsAgo] propiedad esto se referirá automáticamente al objeto de tipo de precios correcto. Esto es un tema avanzado.*

Referirse a Datos de Indicador NinjaScript incluye una biblioteca de incorporado

indicadores a los cuales usted puede tener acceso. Por favor ver la sección de referencia [de Métodos de Indicador](#) para definiciones claras de cómo tener acceso a cada indicador.

Pueden tener acceso a todos los valores de indicador del modo siguiente:

indicador (*parámetros*) [*intervalo* *barsAgo*]

donde el indicador es el nombre del indicador que usted quiere tener acceso, *parámetros* es cualquier parámetro asociado que el indicador requiera y *barsAgo* es el número de barras que deseamos compensar de la barra corriente.

Como un ejemplo, podríamos escribir una declaración al control si la cotización de cierre corriente es mayor que el 20 período [el valor medio móvil simple](#) como esto:

Cerca [0]> SMA (20) [0];

Si usted quisiera realizar el mismo control, pero sólo comprobar contra un 20 período el valor medio móvil simple de altos precios usted lo escribiría como esto:

Cerca [0]> SMA (Alto, 20) [0];

Usted podría escribir una declaración para ver si un 14 indicador CCI de período se eleva como esto:

CCI (14) [0]> CCI (14) [1];

Valor de un 10 período CCI hace 1 barra = CCI (10) [1]

Por favor examine la referencia de Métodos de Indicador para la sintaxis apropiada para tener acceso a valores de indicador diferentes.

-o-

BarsSinceSession

La definición El número de barras que han pasado desde la sesión comenzó.

El Valor de propiedad Un valor internacional que representa el número de barras pasó.

Propiedad De Indicador de Encargo, Estrategia de Encargo

Barras **de sintaxis**. BarsSinceSession

La información de Acceso adicional pueden tener acceso a Esta propiedad sin una referencia nula se registra en OnBarUpdate () controlador de eventos. Cuando OnBarUpdate () el acontecimiento es provocado, siempre habrá un objeto de Barra que sostiene la propiedad de CurrentBid. Si usted desea tener acceso a esta propiedad en otra parte, compruebe la referencia nula primero. p.ej si (iBarras! = nulo)

p354

www.ninjatrader.com
© NinjaTrader, LLC,

Clase de BoolSeries

Definición

BoolSeries es un tipo especial de la estructura de datos que sostiene una serie de valores de bool y siempre contiene el mismo número de elementos que barras en una carta. Ver la Clase [de DataSeries](#) para la información relacionada.

Creación de Objetos de BoolSeries

De crear un objeto de BoolSeries:

1. Defina una variable ("myBoolSeries" usado en este ejemplo) del tipo BoolSeries que creará que BoolSeries se opone 2. En Inicializar () método, cree un nuevo objeto de BoolSeries y adjudíquelo a la variable "myBoolSeries"

```
#region Variables
    BoolSeries privado myBoolSeries;//Definen una variable de BoolSeries #endregion

//Cree un objeto de BoolSeries y adjudique esto a la variable protegida anula el
vacío Inicializan () {
    myBoolSeries = nuevo BoolSeries (esto);//"esto" se refiere al indicador, o
    estrategia                                     //sí mismo. Esto sincroniza
    BoolSeries
    objeto                                     //a barras de datos históricas}
```

* los objetos de BoolSeries pueden ser usados en la serie suplementaria en la estrategia de instrumento y una multimarca de tiempo. Por favor ver nuestro foro de apoyo sección de muestras de referencia de NinjaScript para más información.

El ajuste de Valores - BoolSeries. Juego () & BoolSeries. Reinicio () Usted puede ponerse valor (enchufe un valor) en un objeto de BoolSeries llamando el Juego () método.

BoolSeries. El juego (bool valor) Ajuste de un valor en un objeto de BoolSeries es automáticamente alineado a la barra corriente evaluada. Esto asegura que los valores a través de todos los objetos de BoolSeries siempre están en la sincronización por el índice [de CurrentBar](#). Las muestras de código siguientes demuestran el cálculo de la variedad de cada barra y almacenaje del valor en un objeto de BoolSeries.

```
protegido anulan OnBarUpdate vacío () {
    //Es el final> que myBoolSeries abierto. Juego ({Cerca [0]> Abierto [0]?
    verdadero: falso);
}
```

BoolSeries. Juego (intervalo barsAgo, bool valor) Usted también puede poner el valor para barras históricas por la inclusión de un valor "de barsAgo" que representa el número de hace unas barras que usted quiere que el valor de bool sea almacenado en.

Llamando el Reinicio () el método es único y puede ser muy potente para el desarrollo de indicador de encargo. Los objetos de BoolSeries pueden sostener valores nulos que simplemente significa que usted no quiere almacenar un valor para la barra corriente. El reinicio () reinicializará el valor de índice corriente al nulo.

La comprobación de Valores Válidos

Es posible que usted pueda usar un objeto de BoolSeries, pero decidir no poner un valor para una barra específica. Internamente, un valor falso realmente existe sin embargo, si usted quisiera comprobar para ver si era un valor válido que usted pone, usted puede comprobar el método siguiente.

BoolSeries. ContainsValue (intervalo *barsAgo*) Vuelta un valor verdadero o falso.

La adquisición Valora Usted puede tener acceso a valores de objeto de BoolSeries usando la sintaxis **BoolSeries [intervalo *barsAgo*]** donde *barsAgo* representa el valor de datos n número de hace unas barras.

```
protegido anulan OnBarUpdate vacío () {
    //Letras la Letra de valor de barra corriente y últimas ("Los valores son" +
    myBoolSeries [0] + "" + myBoolSeries [1]);
}
```

-0-

CalculateOnBarClose

Definición De ser verdadero, [OnBarUpdate \(\)](#) es llamado en el final de cada barra por otra parte es llamado en cada marca de entrada. Esta propiedad SÓLO debería ser puesta en [Inicializar \(\)](#) método y ser la última declaración dentro de aquel método.

- Cuando los indicadores o las estrategias corren en datos históricos, OnBarUpdate () sólo es llamado en el final de cada barra histórica aun si esta propiedad es puesta al falso. Esto es debido a que con un conjunto de datos histórico, sólo el OHLCVT de la barra es conocido y no cada marca que arregló la barra.
- La propiedad de CalculateOnBarClose de indicadores introducidos dentro de una estrategia es overridden por la propiedad de CalculateOnBarClose de la estrategia

La propiedad Valora Este propiedad vuelta **verdaderas** si el valor es calculado según el final de cada barra; por otra parte, **falso**. Juego de falta a **falso**.

Sintaxis CalculateOnBarClose

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
//Calcule según el final de cada barra CalculateOnBarClose = verdadero;
```

Puntas 1. El cálculo de indicadores o sistemas para cada marca de entrada puede ser la CPU intensiva. Sólo calcule indicadores según cada marca de entrada si usted tiene un requisito para calcularla barra intra.

-o-

Cerca

p356

www.ninjatrader.com

© NinjaTrader, LLC,

2005

Definición

Una colección de la barra histórica cierra precios.

El Valor de propiedad `DataSet` escribe a máquina el objeto. Teniendo acceso a esta propiedad vía un valor de índice `[el intervalo barsAgo]` devuelve un doble valor que representa el precio de la barra referida.

Sintaxis Final Cercano `[intervalo barsAgo]`

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
//El método de OnBarUpdate protegido anula OnBarUpdate vacío () {  
    //Los controles si el final corriente es mayor que la barra previa cerca si (Cerca  
    [0]> Cerca [1])  
        Letra ("Teníamos un día");  
}
```

-o-

Finales

La definición Sostiene una serie de objetos `de DataSet` sosteniendo la barra histórica precios cercanos. Un objeto de `DataSet` es añadido a esta serie llamando `Añadir ()` método en una Estrategia de Encargo. Su objetivo es proporcionar el acceso a las cotizaciones de cierre de todos los objetos de Barras en la estrategia de multimarca de tiempo o un multiinstrumento.

Valor de propiedad Una serie de objetos de `DataSet`.

Finales de sintaxis `[intervalo barSeriesIndex]` `[intervalo barsAgo]`

Propiedad De Estrategia de Encargo

Ejemplos

p357

www.ninjatrader.co

m

© NinjaTrader, LLC,
2005

```
protegido anulan el vacío Inicializan () {  
    //Añade que unas Barras de 5 minutos se oponen a la estrategia y es  
    automáticamente  
    adjudicado  
    //un índice de objeto de Barras de 1 desde los datos primarios la estrategia  
    es dirigido  
    contra  
    //puesto por el UI toma el índice de 0. Añada (Instrumento, PeriodType. Minuto,  
    5);  
}  
  
protegido anulan OnBarUpdate vacío () {  
    //Compara el precio cercano de la barra primaria con el final de la barra de  
    5 minutos  
    precio  
    si (Finales [0] [0] > Finales [1] [0])  
        La letra ("El precio cercano de la barra corriente es mayor");  
}
```

-o-

Conde

Definición El número total de barras o funciones de datos.

Valor de propiedad Un valor [internacional](#) que representa el número total de barras.

Sintaxis Conde

Propiedad De Indicador de Encargo, Estrategia de Encargo

Puntas 1. El valor [de CurrentBar](#) es garantizado ser \leq el Conde - 1. Esto es debido a la

arquitectura multiteniente de NinjaTrader, el valor de Conde puede tener barras adicionales ya que las marcas en vuelo entran al sistema.

-o-

CountIf ()

La definición cuenta el número de acontecimientos de la condición de prueba durante el período de mirada hacia atrás expresado en barras. **Este método no trabajará en estrategias de multimarca de tiempo o multiinstrumento.**

El Valor de retorno de método Un valor **internacional** que representa el número de acontecimientos encontrado

Sintaxis CountIf (Condición de condición, intervalo lookBackPeriod)

La sintaxis para *la condición* debe escrito en la manera siguiente:

el delegado {devuelve su condición verdadera/falsa;}

Método De Indicador de Encargo, Estrategia de Encargo

p358

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Condición **de parámetros** Una
expresión verdadera/falsa lookBackPeriod Número de barras para mirar
hacia atrás para comprobar la condición de prueba

Ejemplos

```
//Si en las 10 últimas barras hemos tenido 8 las barras entonces van mucho  
tiempo si (CountIf (delegado {vuelven Cerca [0]> Abierto [0];}, 10)> 8)  
EnterLong ();
```

-o-

CrossAbove ()

Controles **de definición** de una cruz encima de condición durante el período de mirada hacia atrás de barra especificado.

El Valor de retorno de método Este método vueltas **verdaderas** si una cruz encima de la condición ocurriera; por otra parte, **falso**.

Sintaxis CrossAbove (IDataSeries series1, doble el valor, el intervalo lookBackPeriod) CrossAbove (IDataSeries series1, IDataSeries series2, intervalo lookBackPeriod)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

valor Cualquier doble valor series1 & series2 Cualesquiera DataSeries escriben a máquina el objeto, como un indicador, Cerca, Alto, Bajo etc. lookBackPeriod Número de barras atrás para comprobar la cruz encima de condición

Ejemplos

```
//Vaya corto si CCI se cruzara encima 250 dentro de la última barra si (CrossAbove
(CCI (14), 250, 1))
    EnterShort ();

//Vaya mucho tiempo si 10 Acuerdo Monetario Europeo se cruza encima de 20 Acuerdo
Monetario Europeo dentro de la última barra si (CrossAbove (Acuerdo Monetario
Europeo (10), Acuerdo Monetario Europeo (20), 1))
    EnterLong ();

//Vaya mucho tiempo tenemos un barra y las 10 cruces de Acuerdo Monetario Europeo
encima de 20 Acuerdo Monetario Europeo dentro de las 5 últimas barras si (Cerca [0]>
Abierto [0] && CrossAbove (Acuerdo Monetario Europeo (10), Acuerdo Monetario
Europeo (20), 5))
    EnterLong ();
```

-0-

CrossBelow ()

Controles **de definición** de una cruz debajo de condición durante el período de mirada hacia atrás de barra especificado.

El Valor de retorno de método Este método vueltas **verdaderas** si una cruz encima de la condición ocurriera; por otra parte, **falso**.

p359

www.ninjatrader.com
© NinjaTrader, LLC,

2005

Sintaxis

CrossBelow (IDataSeries series1, doble el valor, el intervalo
lookBackPeriod) CrossBelow (IDataSeries series1, IDataSeries series2,
intervalo lookBackPeriod)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

valor Cualquier doble valor series1 & series2 Cualesquiera DataSeries escriben a máquina el objeto, como un indicador, Cerca, Alto, Bajo etc. lookBackPeriod Número de barras atrás para comprobar la cruz encima de condición

Ejemplos

```
//Vaya mucho tiempo si CCI se cruzara abajo-250 dentro de la última barra si
(CrossBelow (CCI (14),-250, 1))
    EnterLong ();

//Vaya corto si 10 Acuerdo Monetario Europeo se cruza debajo de 20 Acuerdo Monetario
Europeo dentro de la última barra si (CrossBelow (Acuerdo Monetario Europeo (10),
Acuerdo Monetario Europeo (20), 1))
    EnterShort ();

//Vaya corto tenemos un abajo barra y las 10 cruces de Acuerdo Monetario Europeo
debajo de 20 Acuerdo Monetario Europeo dentro de las 5 últimas barras si (Cerca [0]
<Abierto [0] && CrossBelow (Acuerdo Monetario Europeo (10), Acuerdo Monetario
Europeo (20), 5))
    EnterShort ();
```

-0-

CurrentBar

La definición Un número que representa la barra corriente en unas Barras objeto a que OnBarUpdate () método en un indicador o estrategia trate actualmente. Por ejemplo, si una carta tiene 100 barras de datos, la primera barra de la carta (se marchó la mayor parte de barra) será el número 0 (cero) y cada barra subsecuente de la izquierda a la derecha es incrementada por 1.

La propiedad Valora un valor [internacional](#) que representa la barra corriente.

Sintaxis CurrentBar

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
//El método de OnBarUpdate protegido anula OnBarUpdate vacío () {
    //Controles para asegurarnos tenemos al menos 20 o más barras si (CurrentBar
    <20)
        vuelva;

    //El cálculo de lógica de indicador cifra...
}
```

-0-

p360

www.ninjatrader.com
© NinjaTrader, LLC,

2005

Clase de DataSeries

Definición

DataSeries es un tipo especial de la estructura de datos que sostiene una serie de dobles valores y siempre contiene el mismo número de elementos que barras en una carta. Si usted tiene 200 barras cargadas en su carta por un valor medio móvil trazado, el valor medio móvil

sí mismo sostiene un objeto de **DataSeries** con 200 valores históricos de datos, un para cada barra. Los objetos de **DataSeries** pueden ser usados como datos de entrada para todos [los métodos de indicador](#). La clase de **DataSeries** pone en práctica el interfaz [de IDataSeries](#).

Creando Objetos de **DataSeries** creando indicadores de encargo, los objetos de **DataSeries** son automáticamente creados para usted llamando [Añadir \(\)](#) método y pueden ser referidos posteriormente por la propiedad [de Valores](#) y/o el [Valor](#). Sin embargo, usted puede tener un requisito para crear un objeto de **DataSeries** de almacenar valores que son la parte de un cálculo de valor de indicador total. Esto puede ser hecho dentro de un indicador de encargo o estrategia.

Crear un objeto de **DataSeries**:

1. Defina una variable ("myDataSeries" usado en este ejemplo) del tipo **DataSeries** que creará que **DataSeries** se opone 2. En Inicializar () método, cree un nuevo objeto de **DataSeries** y adjudíquelo a la variable "myDataSeries"

```
#region Variables
    DataSeries privado myDataSeries;//Definen una variable de DataSeries #endregion

//Cree un objeto de DataSeries y adjudique esto a la variable protegido anula el vacío Inicializan () {
    myDataSeries = nuevo DataSeries (esto);//"esto" se refiere al indicador, o
    estrategia
                                //sí mismo. Esto sincroniza
DataSeries
objeto
                                //a barras de datos históricas}
```

* los objetos de **DataSeries** pueden ser usados en la serie suplementaria en la estrategia de instrumento y una multimarca de tiempo. Por favor ver nuestro foro de apoyo sección de muestras de referencia de NinjaScript para más información.

El ajuste de Valores - **DataSeries. Juego () & **DataSeries**. Reinicio ()** Usted puede ponerse valor (enchufe un valor) en un objeto de **DataSeries** llamando el **Juego ()** método.

****DataSeries**. El juego (doble **valor**)** Ajuste de un valor en un objeto de **DataSeries** es automáticamente alineado a la barra corriente evaluada. Esto asegura que los valores a través de todos los objetos de **DataSeries** siempre están en la sincronización por el índice [de CurrentBar](#). Las muestras de código siguientes demuestran el cálculo de la variedad de cada barra y almacenaje del valor en un objeto de **DataSeries**.

```
protegido anulan OnBarUpdate vacío () {
    //Calcule la variedad de la barra corriente y ponga el valor myDataSeries. Juego
    (Alto [0] - Bajo [0]);
}
```

****DataSeries**. Juego (intervalo **barsAgo**, doble el **valor**)** Usted también puede poner el valor para barras históricas por la inclusión de un valor "de barsAgo" que representa el número de hace unas barras que usted quiere que el doble valor sea almacenado en.

Llamando el Reinicio () el método es único y puede ser muy potente para el desarrollo de indicador de encargo. Los objetos de `DataSet` pueden sostener valores nulos que simplemente significa que usted no quiere almacenar un valor para la barra corriente. Matemáticamente, usted puede adjudicar correctamente un valor del cero sin embargo si `DataSet` fuera `DataSet` primario de un indicador cuyos valores serían usados para [la conspiración](#), usted no puede querer un valor cero trazado. Sentido, usted quiere un valor cero para cálculos apropiados, pero no un valor cero para la visualización de carta. El Reinicio () el método permite que usted reinicialice el valor de `DataSet` de la barra corriente a un cero con objetivos de cálculo pero NinjaScript no haría caso de este valor cuando esto lo trazó en una carta.

La comprobación de Valores Válidos es posible que usted pueda usar un objeto de `DataSet`, pero decidir no poner un valor para una barra específica. Internamente, un valor falso realmente existe que es la barra corriente cerca sin embargo, si usted quisiera comprobar para ver si era un valor válido que usted pone, usted puede comprobar el método siguiente.

`DataSet.ContainsValue (intervalo barsAgo)` Vueltas un valor verdadero o falso.

La adquisición Valora Usted puede tener acceso a valores de objeto de `DataSet` usando la sintaxis `DataSet [intervalo barsAgo]` donde `barsAgo` representa el valor de datos n número de hace unas barras.

```
protegido anulan OnBarUpdate vacío () {  
    //Imprime el valor de barra corriente y último  
  
    si (CurrentBar > 0)  
        La letra ("Los valores son" + myDataSet [0] + "" + myDataSet [1]);}
```

Los métodos que Aceptan DataSet como métodos de indicador Arguments All aceptan objetos de `DataSet` como argumentos. Llevando de los ejemplos previos, vaya a la copia impresa del 10 período valor medio móvil simple de la variedad.

```
protegido anulan OnBarUpdate vacío () {  
    //Calcule la variedad de la barra corriente y ponga el valor myDataSet. Juego  
    (Alto [0] - Bajo [0]);  
  
    //Letra el 10 período corriente SMA de la Letra de variedad ("Valor es" + SMA  
    (myDataSet, 10) [0]);  
}
```

-o-

Clase de `DateTimeSeries`

La definición DateTimeSeries es un tipo especial de la estructura de datos que sostiene una serie de valores de DateTime y siempre contiene el mismo número de elementos que barras en una carta. Ver la Clase [de DataSeries](#) para la información relacionada.

Creación de Objetos de DateTimeSeries

p362

www.ninjatrader.com
© NinjaTrader, LLC,

2005

Crear un objeto de DateTimeSeries:

1. Defina una variable ("myDateTimeSeries" usado en este ejemplo) del tipo DateTimeSeries que creará que DateTimeSeries se opone 2. En Inicializar () método, cree un nuevo objeto de DateTimeSeries y adjudíquelo a la variable "myDateTimeSeries"

```
#region Variables
    DateTimeSeries privado myDateTimeSeries;//Definen una variable de DateTimeSeries
#endregion

//Cree un objeto de DateTimeSeries y adjudique esto a la variable protegida anula
el vacío Inicializan () {
    myDateTimeSeries = nuevo DateTimeSeries (esto);//"esto" se refiere al
    indicador, o estrategia
    Objeto de DateTimeSeries
    //sí mismo. Esto sincroniza el
    //a barras de datos históricas}
```

* los objetos de DateTimeSeries pueden ser usados en la serie suplementaria en la estrategia de instrumento y una multimarca de tiempo. Por favor ver nuestro foro de apoyo sección de muestras de referencia de NinjaScript para más información.

El ajuste de Valores - DateTimeSeries. Juego () & DateTimeSeries. Reinicio () Usted puede ponerse valor (enchufe un valor) en un objeto de DateTimeSeries llamando el Juego () método.

DateTimeSeries. El juego (valor de DateTime) Ajuste de un valor en un objeto de DateTimeSeries es automáticamente alineado a la barra corriente evaluada. Esto asegura que los valores a través de todos los objetos de DateTimeSeries siempre están en la sincronización por el índice [de CurrentBar](#). Las muestras de código siguientes demuestran el cálculo de la variedad de cada barra y almacenaje del valor en un objeto de DateTimeSeries.

```
protegido anulan OnBarUpdate vacío () {
    //Almacene el tiempo corriente myDateTimeSeries. Juego (DateTime. Ahora);
}
```

DateTimeSeries. Juego (intervalo barsAgo, valor de DateTime) Usted también puede poner el valor para barras históricas por la inclusión de un valor "de barsAgo" que representa el número de hace unas barras que usted quiere que el valor de DateTime sea almacenado en.

Llamando el Reinicio () el método es único y puede ser muy potente para el desarrollo de

indicador de encargo. Los objetos de DateTimeSeries pueden sostener valores nulos que simplemente significa que usted no quiere almacenar un valor para la barra corriente. El reinicio () reinicializará el valor de índice corriente al nulo.

La comprobación de Valores Válidos es posible que usted pueda usar un objeto de DateTimeSeries, pero decidir no poner un valor para una barra específica. Internamente, un valor falso realmente existe sin embargo, si usted quisiera comprobar para ver si era un valor válido que usted pone, usted puede comprobar el método siguiente.

DateTimeSeries. ContainsValue (intervalo barsAgo) Vuelta un valor verdadero o falso.

p363

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Adquisición de Valores

Usted puede tener acceso a valores de objeto de DateTimeSeries usando la sintaxis **DateTimeSeries [intervalo barsAgo]** donde barsAgo representa el valor de datos n número de hace unas barras.

```
protegido anulan OnBarUpdate vacío () {  
    //Letras la Letra de valor de barra corriente y últimas ("Los valores son" +  
    myDateTimeSeries [0] + "" +  
    myDateTimeSeries [1]);}
```

-o-

DayOfWeek

Tienen acceso **a la definición** El día de la semana de la barra corriente por la propiedad [de Tiempo](#).

Valor de propiedad Un tipo de DayOfWeek, como: DayOfWeek. El lunes DayOfWeek. El martes DayOfWeek. El miércoles DayOfWeek. El jueves DayOfWeek. El viernes DayOfWeek. El sábado DayOfWeek. El domingo

Tiempo de sintaxis [intervalo barsAgo].DayOfWeek

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    //Si su lunes... no comercia si (Tiempo [0].DayOfWeek == DayOfWeek. El lunes)  
    vuelva;  
}
```

-o-

Disponga ()

Definición Para programadores avanzados: Usted puede anular Disponer () método de limpiar todos los recursos poseídos por el indicador de encargo o estrategia. Es **crítico que usted llame la base. Disponga () el método anulando Disponer () método.**

Método De Indicador de Encargo, Estrategia de Encargo

Ejemplos

p364
www.ninjatrader.co
m
© NinjaTrader, LLC, 2005

```
el público anula el vacío Disponen () {  
    //Limpie sus recursos aquí basan.  
    Disponga ();  
}
```

-o-

Caída ()

Los Controles **de definición** de una condición decreciente que es verdad cuando el valor corriente es menos que el valor de hace 1 barra.

El Valor de retorno de método Este método vueltas **verdaderas** si una condición decreciente está presente; por otra parte, **falso**.

Caída de sintaxis (*serie de IDataSeries*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

serie Cualesquiera DataSeries escriben a máquina el objeto, como un indicador, Cerca, Alto, Bajo etc.

Ejemplos

```
//Si el 20 período SMA se cae (en downtrend) va corto si (Cayéndose (SMA (20))  
EnterShort ();
```

-o-

FirstBarOfSession

La definición Consigue un valor que indica si la barra es la primera barra de una sesión.

La propiedad Valora Este propiedad vueltas **verdaderas** si la barra es la primera barra de una sesión; por otra parte, **falso**.

Barras **de sintaxis**. FirstBarOfSession

Propiedad De Indicador de Encargo, Estrategia de Encargo

La información de Acceso adicional pueden tener acceso a Esta propiedad sin una referencia nula se registra en OnBarUpdate () controlador de eventos. Cuando OnBarUpdate () el acontecimiento es provocado, siempre habrá un objeto de Barra que sostiene la propiedad de FirstBarOfSession. Si usted desea tener acceso a esta propiedad en otra parte, compruebe la referencia nula primero. p.ej si (iBarras! = nulo)

-0-

FirstTickOfBar

p365

www.ninjatrader.com
© NinjaTrader, LLC,

2005

Definición

Indica si la marca de entrada es la primera marca de una nueva barra. Esta propiedad sólo es de valor en estrategias que dirigen la marca por la marca que es cuando la propiedad [de CalculateOnBarClose](#) es puesta al falso.

La propiedad Valora Este propiedad vueltas **verdaderas** si la marca de entrada es la primera marca de una nueva barra; por otra parte, **falso**.

Sintaxis FirstTickOfBar

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
//En una marca por la estrategia de marca de único camino usted sabe cuando una barra
está cerrada es cuando//FirstTickOfBar es verdad. protegido anulan OnBarUpdate vacío
() {
    //Sólo las señales de entrada de proceso en una barra por la base de barra (no
    hacen tictac por la marca) si (FirstTickOfBar) {
        si (CCI (20) [1] <-250)
            EnterLong ();

        vuelva;}

    //La salida de proceso señala la marca por la marca si (CCI (20) [0]> 250)
        ExitLong ();
}
```

-0-

Clase de FloatSeries

La definición FloatSeries es un tipo especial de la estructura de datos que sostiene una serie de valores de flotador y siempre contiene el mismo número de elementos que barras en una carta. Ver la Clase [de DataSeries](#) para la información relacionada.

Creación de Objetos de FloatSeries De crear un objeto de FloatSeries:

1. Defina una variable ("myFloatSeries" usado en este ejemplo) del tipo FloatSeries que creará que FloatSeries se opone 2. En Inicializar () método, cree un nuevo objeto de FloatSeries y adjudíquelo a la variable "myFloatSeries"

p366

www.ninjatrader.co

m

© NinjaTrader, LLC,

2005

```
#region Variables
    FloatSeries privado myFloatSeries; //Definen una variable de FloatSeries
#endregion

//Cree un objeto de FloatSeries y adjudique esto a la variable protegida anula el
vacío Inicializan () {
    myFloatSeries = nuevo FloatSeries (esto); // "esto" se refiere al indicador,
    o estrategia
    objeto
    //sí mismo. Esto sincroniza IntSeries
    //a barras de datos históricas}
```

* los objetos de FloatSeries pueden ser usados en la serie suplementaria en la estrategia de instrumento y una multimarca de tiempo. Por favor ver nuestro foro de apoyo sección de muestras de referencia de NinjaScript para más información.

El ajuste de Valores - FloatSeries. Juego () & FloatSeries. Reinicio () Usted puede ponerse valor (enchufe un valor) en un objeto de FloatSeries llamando el Juego () método.

FloatSeries. El juego (valor de flotador) Ajuste de un valor en un objeto de FloatSeries es automáticamente alineado a la barra corriente evaluada. Esto asegura que los valores a través de todos los objetos de FloatSeries siempre están en la sincronización por el índice [de CurrentBar](#). Las muestras de código siguientes demuestran el cálculo de la variedad de cada barra y almacenaje del valor en un objeto de FloatSeries.

```

protegido anulan OnBarUpdate vacío () {
    ponga a flote la variedad = (flotador) Alto [0] - Bajo [0]; myFloatSeries. Juego
    (variedad);
}

```

FloatSeries. Juego (intervalo *barsAgo*, bool *valor*) Usted también puede poner el valor para barras históricas por la inclusión de un valor "de barsAgo" que representa el número de hace unas barras que usted quiere que el valor de bool sea almacenado en.

Llamando el Reinicio () el método es único y puede ser muy potente para el desarrollo de indicador de encargo. Los objetos de FloatSeries pueden sostener valores nulos que simplemente significa que usted no quiere almacenar un valor para la barra corriente. El reinicio () reinicializará el valor de índice corriente al nulo.

La comprobación de Valores Válidos es posible que usted pueda usar un objeto de FloatSeries, pero decidir no poner un valor para una barra específica. Internamente, un valor falso realmente existe sin embargo, si usted quisiera comprobar para ver si era un valor válido que usted pone, usted puede comprobar el método siguiente.

FloatSeries. ContainsValue (intervalo *barsAgo*) Vueltas un valor verdadero o falso.

La adquisición Valora Usted puede tener acceso a valores de objeto de IntSeries usando la sintaxis **IntSeries [intervalo *barsAgo*]** donde barsAgo representa el valor de datos n número de hace unas barras.

p367

www.ninjatrader.co

m

© NinjaTrader, LLC, 2005

```

protegido anulan OnBarUpdate vacío () {
    //Letras la Letra de valor de barra corriente y últimas ("Los valores son" +
    myFloatSeries [0] + "" + myFloatSeries [1]);
}

```

-o-

GetBar ()

Las Vueltas **de definición** el número de hace unas barras comenzando con la 1ra barra del más viejo al más nuevo que corresponde al sello de tiempo basado en el parámetro de **DateTime** <<http://msdn2.microsoft.com/en-us/library/system.datetime.aspx>> **pasaron en**. Si el sello de tiempo pasara en es más viejo que la 1ra barra, el valor volvió refleja el número de hace unas barras para la 1ra barra de la serie. Si el sello de tiempo es más nuevo que la última barra, un valor del cero es devuelto.

El Valor de retorno de método Un valor **internacional** que representa el número de hace unas barras.

Sintaxis `GetBar (tiempo de DateTime) GetBar (intervalo barSeriesIndex, tiempo de DateTime)`

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

tiempo Un caso de tiempo. barSeriesIndex El valor de índice de barras se opone en una estrategia de multiserie

Ejemplos

```
//Compruebe que sus las 9:45 pasadas
si (ToTime (Tiempo [0])> = ToTime (9, 45, 00)) {
    //Cuenta hace las barras valoran por la barra de 9:00 para el intervalo de
    día corriente barsAgo = GetBar (nuevo DateTime (2006, 12, 18, 9, 0, 0));

    //La copia impresa de la Letra de cotización de cierre de barra de 9:00 ("El
    precio cercano en la barra de 9:00 era:" +
    Cerca [barsAgo].ToString ());}
```

-O-

GetCurrentAsk ()

Las Vueltas **de definición** la corriente preguntan al precio.

El Valor de retorno de método Un **doble** valor que representa la corriente pregunta al precio. Note: Cuando tenido acceso durante backtest histórico, el precio cercano de la barra evaluada es subsituted.

Sintaxis `GetCurrentAsk () GetCurrentAsk (intervalo barSeriesIndex)`

Método De

p368

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Indicador de encargo, Estrategia de Encargo

-O-

GetCurrentAskVolume ()

Las Vueltas **de definición** la corriente preguntan al volumen.

El Valor de retorno de método Un valor **internacional** que representa la corriente pregunta al volumen. Note: Cuando tenido acceso durante backtest histórico, el volumen de la barra evaluada es subsituted.

Sintaxis `GetCurrentAskVolume () GetCurrentAskVolume (intervalo barSeriesIndex)`

Método De Indicador de Encargo, Estrategia de Encargo

-o-

GetCurrentBid ()

Vueltas **de definición** el precio de puja actual.

Valor de retorno de método Un **doble** valor que representa el precio de puja actual. Note: Cuando tenido acceso durante backtest histórico, el precio cercano de la barra evaluada es substituted.

Sintaxis GetCurrentBid () **GetCurrentBid (intervalo barSeriesIndex)**

Método De Indicador de Encargo, Estrategia de Encargo

-o-

GetCurrentBidVolume ()

Vueltas **de definición** el volumen de puja actual.

Valor de retorno de método Un valor **internacional** que representa el volumen de puja actual. Note: Cuando tenido acceso durante backtest histórico, el volumen de la barra evaluada es substituted.

Sintaxis GetCurrentBidVolume () **GetCurrentBidVolume (intervalo barSeriesIndex)**

Método De Indicador de Encargo, Estrategia de Encargo

-o-

GetMedian ()

Vueltas **de definición** el valor mediano de la serie especificada durante el período de mirada hacia atrás especificado. Esto

p369

www.ninjatrader.com
© NinjaTrader, LLC, 2005

el método clasificará los valores del período de mirada hacia atrás especificado en orden ascendente y devolverá el valor medio. Si un número par es pasado en, el promedio de los dos valores medios en los valores clasificados será devuelto.

Valor de retorno de método Un **doble** valor que representa el precio mediano.

Sintaxis GetMedian (serie de IDataSeries, intervalo lookBackPeriod)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

serie Cualesquiera DataSeries escriben a máquina el objeto, como un indicador, Cerca,

Alto, Bajo etc. lookBackPeriod Número de barras atrás para incluir en el cálculo

Ejemplos

```
//Imprima el precio mediano de los 10 últimos precios abiertos
si (GetMedian (Abierto, 10))
    La letra ("La mediana de los 10 últimos precios abiertos es:" + GetMedian
(Abierto,
10). ToString ());
```

-0-

GetSessionBar ()

La definición Devuelve un objeto de Barra que representa una sesión a cuyas propiedades para el abierto, alto, bajo, cerca, tiempo y volumen pueden tener acceso. Esta barra es una barra virtual construida lejos de la serie de datos subyacente. En algunos casos, la barra de sesión puede ser usada para representar una barra diaria. Ya que esta barra es virtual, sus valores de propiedad son calculados para el tiempo de sesión sólo y no representan el día actual. Usted debe comprobar una referencia nula ya que nulo es devuelto si no hay datos de un día suficientes para construir una barra de sesión.

!!!! Este método SÓLO debería ser usado para tener acceso a datos de sesión previos. Para tener acceso a datos de sesión corrientes, use [CurrentDayOHL \(\)](#) método.

Valor de retorno de método Véase abajo.

La sintaxis Las propiedades debajo de la vuelta [dobla](#) valores: [Barras. GetSessionBar \(intervalo sessionsAgo\).Open](#) [Barras. GetSessionBar \(intervalo sessionsAgo\).High](#) [Barras. GetSessionBar \(intervalo sessionsAgo\).Low](#) [Barras. GetSessionBar \(intervalo sessionsAgo\).Close](#)

La propiedad debajo de vueltas una estructura [de DateTime](#):

<<http://msdn2.microsoft.com/en-us/library/system.datetime.aspx>> [Bars.GetSessionBar \(intervalo sessionsAgo\).Time](#)

La propiedad debajo de vueltas un valor [internacional](#): [Barras. GetSessionBar \(intervalo sessionsAgo\).Volume](#)

Método De Indicador de Encargo, Estrategia de Encargo

p370

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Parámetros

sessionsAgo Número de hace unas sesiones

Ejemplos

```
//Imprima la sesión previa cerca
si (Barras. ¿GetSessionBar (1) != nulo)
    La letra ("El final de la sesión previa es:" +
Barras. GetSessionBar (1).Close. ToString ());
```

-o-

HighestBar ()

Las Vueltas **de definición** el número de hace unas barras el valor de precios más alto ocurrieron para el período lookback. Si hay varios sucesos dentro del período lookback donde la barra era la barra más alta el suceso más reciente es devuelto.

El Valor de retorno de método Un valor **internacional** que representa el número de hace unas barras.

Sintaxis HighestBar (*serie de IDataSeries, intervalo lookBackPeriod*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

serie Cualesquiera DataSeries escriben a máquina el objeto, como un indicador, Cerca, Alto, Bajo etc. lookBackPeriod El número de barras para incluir en el cálculo

Ejemplos

```
//Imprime el número de hace unas barras para el final más alto de la Letra de sesión
corriente (HighestBar (Cerca, Barras. BarsSinceSession - 1).ToString ());
```

-o-

Alto

Definición Una colección de altos precios de barra históricos.

El Valor de propiedad DataSeries escribe a máquina el objeto. Teniendo acceso a esta propiedad vía un valor de índice **[el intervalo barsAgo]** devuelve un doble valor que representa el precio de la barra referida.

Sintaxis Alto Alto **[intervalo barsAgo]**

Propiedad De Indicador de Encargo, Estrategia de Encargo

p371

www.ninjatrader.com
© NinjaTrader, LLC,

Ejemplos

```
//Método de OnBarUpdate
protegido anulan OnBarUpdate vacío () {
    //Asegúrese que tenemos al menos 20 barras si (CurrentBar <20)
    vuelva;

    //Controles de máximos más altos si (Alto [0]> Alto [1] && Alto [1]> Alto [2])
    Letra ("Dos máximos más altos sucesivos");

    //Consigue el valor corriente de un 20 período SMA de altos precios doble valor
    = SMA (Alto, 20) [0]; la Letra ("El valor es" + valor. ToString ());
}
```

-0-

Máximos

La definición Sostiene una serie de objetos [de DataSeries](#) sosteniendo altos precios de barra históricos. Un objeto de DataSeries es añadido a esta serie llamando [Añadir \(\)](#) método en una Estrategia de Encargo. Su objetivo es proporcionar el acceso a los altos precios de todos los objetos de Barras en la estrategia de multimarca de tiempo o un multiinstrumento.

Valor de propiedad Una serie de objetos de DataSeries.

Máximos de sintaxis [intervalo *barSeriesIndex*] [intervalo *barsAgo*]

Propiedad De Estrategia de Encargo

Ejemplos

```
protegido anulan el vacío Inicializan () {
    //Añade que unas Barras de 5 minutos se oponen a la estrategia y es
    automáticamente
    adjudicado
    //un índice de objeto de Barras de 1 desde los datos primarios la estrategia
    es dirigido
    contra
    //puesto por el UI toma el índice de 0. Añada (Instrumento, PeriodType. Minuto,
    5);
}

protegido anulan OnBarUpdate vacío () {
    //Compara el alto precio de la barra primaria con el alto precio de la barra
    de 5 minutos si (Máximos [0] [0]> Máximos [1] [0])
    La letra ("El alto precio de la barra corriente es mayor");
}
```

-0-

Histórico

Definición

Indica si el objeto de NinjaScript corre en datos históricos. Con una estrategia, dirigiendo un backtest, esta propiedad siempre volverá verdadero. Dirigiendo una estrategia viva, esta propiedad será verdad hasta que la estrategia comience a tratar datos de tiempo real. La propiedad puede ser de valor para muchas aplicaciones diferentes.

1. Eliminando datos históricos para estrategias de tiempo real que no requieren que ningunos datos históricos inicialicen.

La propiedad Valora Este propiedad vueltas **verdaderas** si la estrategia corre en datos históricos; por otra parte, **falso**.

Sintaxis Histórica

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    //Sólo dirigido en datos de tiempo real de Ser (histórico)  
    vuelva;  
}
```

-0-

IDataSeries

IDataSeries **de definición** es un interfaz que es puesto en práctica por todas las clases de NinjaScript que manejan datos históricos, como [DataSeries](#), Indicadores y otras clases. La importancia de esto se reduce a un uso práctico que le provee de los medios de escribir un método que tiene la flexibilidad en los tipos de series de datos de precios que esto puede aceptar. Especificando un parámetro de tipo IDataSeries, usted puede pasar entonces en una serie de cotizaciones de cierre, un indicador o un usuario definieron la serie de datos.

El código de muestra abajo demuestra un método llamado DoubleTheValue que acepta cualquier objeto que ponga en práctica el interfaz de IDataSeries como un parámetro. Este método es usado entonces dos veces, la primera vez pasando en una serie de cotizaciones de cierre y la segunda vez pasando en un 20 período valor medio móvil simple.

```
doble DoubleTheValue privado (IDataSeries priceData) {  
    devuelve priceData [0] * 2;}  
  
protegido anulan OnBarUpdate vacío () {  
    Letra (DoubleTheValue (Cerca)); Letra (DoubleTheValue (SMA (20)));  
}
```

-0-

Inicialice ()

Definición

Inicializar () método es llamado una vez comenzando una estrategia. Este método puede ser usado para configurar propiedades de estrategia e indicador.

Por ejemplo:

- Estrategias - Llamada [Añadir \(\)](#) método de añadir indicadores a cartas y añadir Barras adicionales
- los objetos a la estrategia
- Indicadores - propiedades de configuración de Juego, como [el Revestimiento](#) y [PaintPriceMarkers](#).
- Aunque este método sea llamado una vez en el principio de un indicador o estrategia, hay tiempos que este método también sea llamado (Apertura de un diálogo de indicador) cuando usted no puede esperarlo a. **PARA PROGRAMADORES AVANZADOS, NO USE ESTE MÉTODO DE DIRIGIR RUTINAS DE ENCARGO, COMO LA APERTURA DE FORMAS, HACIENDO LLAMADAS EXTERNAS, COMO UN CONTROL DE LICENCIA.**

El Valor de retorno de método Este método no devuelve un valor.

La sintaxis Ve el ejemplo abajo. El indicador NinjaScript y los magos de estrategia automáticamente generan la sintaxis de método para usted.

Método De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
protegido anulan el vacío Inicializan () {
    //Añade que unas Barras de 5 minutos se oponen a la estrategia y es
    automáticamente
    adjudicado
    //un índice de objeto de Barras de 1 desde los datos primarios la estrategia
    es dirigido
    contra
    //puesto por el UI toma el índice de 0. Añada (Instrumento, PeriodType. Minuto,
    5);
}
```

-O-

Entrada

Definición La introducción de datos principal.

El Valor de propiedad Un tipo de serie de datos objeto a que ponga en práctica el interfaz_

de [IDataSeries](#). Teniendo acceso a esta propiedad vía un valor de índice [[el intervalo barsAgo](#)] devuelve un doble valor que representa el precio de la barra referida.

Entrada de Entrada **de sintaxis** [[intervalo barsAgo](#)]

Propiedad De Indicador de Encargo, Estrategia de Encargo

p374

www.ninjatrader.com
© NinjaTrader, LLC,

2005

Ejemplos

```
//Imprime el valor corriente de la Letra de entrada (Entrada [0].ToString ());
```

Puntas 1. La entrada representa la fuente de datos de entrada del indicador. Si usted pone en práctica un indicador donde PriceTypeSupported es puesto al verdadero, entonces refiriéndose al valor de fuente de datos de entrada, usted querrá referirse **a la Entrada [0]** en vez del Cercano [0]. Esto es porque **la Entrada [0]** devolverá el valor de datos de [PriceType](#) selected.

-0-

Clase de instrumento

Definición Un instrumento tradeable.

Métodos y Propiedades

Vencimiento Ver [referencia](#) FullName Ver [referencia](#)
MasterInstrument. InstrumentTy Ven [referencia](#)
pe
MasterInstrument. Nombre Ver [referencia](#)
MasterInstrument. PointValue Ver [referencia](#)
MasterInstrument. Round2TickSi Ven [referencia](#) Ze

-0-

Instrumento. Vencimiento

Vueltas **de definición** la fecha de caducidad de un contrato de futuros.

Valor de propiedad

<<http://msdn2.microsoft.com/en-us/library/system.datetime.aspx>>A estructura [de DateTime](#) que representa la fecha de caducidad de un contrato de futuros.

Instrumento **de sintaxis**. Vencimiento

Propiedad De Indicador de Encargo, Estrategia de Encargo

La información de Acceso adicional pueden tener acceso a Esta propiedad sin una referencia nula se registra en OnBarUpdate () controlador de eventos. Cuando OnBarUpdate () el acontecimiento es provocado, siempre habrá un objeto de Instrumento. Si usted desea tener acceso a esta propiedad en otra parte, compruebe la referencia nula primero. p.ej si (iInstrumento! = nulo)

-o-

Instrumento. FullName

Vueltas **de definición** el nombre de NinjaTrader lleno de un instrumento. Para el futuro, esto incluiría el

p375

www.ninjatrader.com
© NinjaTrader, LLC, 2005

fecha de caducidad. El junio S&P 500 Emini se contraen el nombre completo es "ES 06-07".

Valor de propiedad Una [cuerda](#) que representa el nombre completo del instrumento.

Instrumento **de sintaxis**. FullName

Propiedad De Indicador de Encargo, Estrategia de Encargo

La información de Acceso adicional pueden tener acceso a Esta propiedad sin una referencia nula se registra en OnBarUpdate () controlador de eventos. Cuando OnBarUpdate () el acontecimiento es provocado, siempre habrá un objeto de Instrumento. Si usted desea tener acceso a esta propiedad en otra parte, compruebe la referencia nula primero. p.ej si (iInstrumento! = nulo)

-o-

Instrumento. MasterInstrument. Compárese ()

La definición Compara dos valores de precios con respecto al Instrumento [TickSize](#) para asegurar la exactitud tratando con matemáticas de punto flotante.

Valor de retorno de método Un valor [internacional](#).

Un valor de "1" es devuelto si price1 es mayor que price2 Un valor de "-1" es devuelto si price1 es menos que price2 Un valor de "0" si price1 es igual a price2

Instrumento **de sintaxis**. MasterInstrument. Compárese ([doble price1](#), [doble price2](#))

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

price1 Un [doble](#) valor que representa un precio price2Un [doble](#) valor que representa un precio

Ejemplos

```
double newPrice = Cerca [0] + Alto [0] + Abierto [0]; si (Instrumento.  
MasterInstrument. Compárese (newPrice, Cerca [1]) == 1)  
    //Haga algo ya que el price1 es mayor que price2
```

-o-

Instrument.MasterInstrument.InstrumentTy

pe

Vueltas **de definición** el tipo de instrumento.

Valor de propiedad

p376

www.ninjatrader.com
© NinjaTrader, LLC, 2005

[InstrumentType](#) que representa el tipo de un instrumento.

Valores posibles son: InstrumentType. Dinero InstrumentType. Futuro InstrumentType. Índice InstrumentType. Opción InstrumentType. Reserva

Sintaxis Instrument.MasterInstrument.InstrumentType

Propiedad De Indicador de Encargo, Estrategia de Encargo

La información de Acceso adicional pueden tener acceso a Esta propiedad sin una referencia nula se registra en OnBarUpdate () controlador de eventos. Cuando OnBarUpdate () el acontecimiento es provocado, siempre habrá un objeto de Instrumento. Si usted desea tener acceso a esta propiedad en otra parte, compruebe la referencia nula primero. p.ej si (iInstrumento! = nulo)

-o-

Instrumento. MasterInstrument. Nombre

Vueltas **de definición** el nombre de base de datos de NinjaTrader de un instrumento. Por ejemplo, "MSFT", "ES", "ER2" etc...

Valor de propiedad Una [cuerda](#) que representa el nombre del instrumento.

Instrumento **de sintaxis**. MasterInstrument. Nombre

Propiedad De Indicador de Encargo, Estrategia de Encargo

La información de Acceso adicional pueden tener acceso a Esta propiedad sin una referencia nula se registra en OnBarUpdate () controlador de eventos. Cuando OnBarUpdate

() el acontecimiento es provocado, siempre habrá un objeto de Instrumento. Si usted desea tener acceso a esta propiedad en otra parte, compruebe la referencia nula primero. p.ej si (iInstrumento! = nulo)

-o-

Instrumento. MasterInstrument. PointValue

Vueltas **de definición** el valor monetario de 1 punto lleno de movimiento. Por ejemplo, 1 punto en el S&P 500 contrato de futuros de Emini (ES) es el USD de 50\$ que es igual al USD de 12,50\$ por marca.

Valor de propiedad Un **doble** valor que representa el valor monetario de 1 punto de movimiento.

Instrumento **de sintaxis**. MasterInstrument. PointValue

Propiedad De

p377

www.ninjatrader.com
© NinjaTrader, LLC,

2005

Indicador de encargo, Estrategia de Encargo

Información de Acceso adicional

Pueden tener acceso a esta propiedad sin una referencia nula se registran en OnBarUpdate () controlador de eventos. Cuando OnBarUpdate () el acontecimiento es provocado, siempre habrá un objeto de Instrumento. Si usted desea tener acceso a esta propiedad en otra parte, compruebe la referencia nula primero. p.ej si (iInstrumento! = nulo)

-o-

Instrument.MasterInstrument.Round2TickSize ()

La definición Devuelve un valor sobre que dobla al valor válido más cercano regularmente divisible la talla de marca del instrumento.

Valor de retorno de método Un **doble** valor.

La sintaxis Instrument.MasterInstrument.Round2TickSize (doblan *el precio*)

Método De Indicador de Encargo, Estrategia de Encargo

Precio **de parámetros** Un **doble** valor que
representa un precio

La información de Acceso adicional pueden tener acceso a Esta propiedad sin una referencia nula se registra en OnBarUpdate () controlador de eventos. Cuando OnBarUpdate () el acontecimiento es provocado, siempre habrá un objeto de Instrumento. Si usted desea

tener acceso a esta propiedad en otra parte, compruebe la referencia nula primero. p.ej si (iInstrumento! = nulo)

-o-

Clase de IntSeries

La definición IntSeries es un tipo especial de la estructura de datos que sostiene una serie de valores internacionales y siempre contiene el mismo número de elementos que barras en una carta. Ver la Clase [de DataSeries](#) para la información relacionada.

Creación de Objetos de IntSeries De crear un objeto de IntSeries:

1. Defina una variable ("myIntSeries" usado en este ejemplo) del tipo IntSeries que creará que IntSeries se opone 2. En Inicializar () método, cree un nuevo objeto de IntSeries y adjudíquelo a la variable "myIntSeries"

2005

p378
www.ninjatrader.co
m
© NinjaTrader, LLC,

```
#region Variables
    IntSeries privado myIntSeries;//Definen una variable de IntSeries #endregion

//Cree un objeto de IntSeries y adjudique esto a la variable protegida anula el vacío
Inicializan () {
    myIntSeries = nuevo IntSeries (esto);//"esto" se refiere al indicador, o
    estrategia                                     //sí mismo. Esto sincroniza IntSeries
    objeto                                     //a barras de datos históricas}
```

* los objetos de IntSeries pueden ser usados en la serie suplementaria en la estrategia de instrumento y una multimarca de tiempo. Por favor ver nuestro foro de apoyo sección de muestras de referencia de NinjaScript para más información.

El ajuste de Valores - IntSeries. Juego () & IntSeries. Reinicio () Usted puede ponerse valor (enchufe un valor) en un objeto de IntSeries llamando el Juego () método.

IntSeries. El juego (valor internacional) Ajuste de un valor en un objeto de IntSeries es automáticamente alineado a la barra corriente evaluada. Esto asegura que los valores a través de todos los objetos de IntSeries siempre están en la sincronización por el índice [de CurrentBar](#). Las muestras de código siguientes demuestran el cálculo de la variedad de cada barra y almacenaje del valor en un objeto de IntSeries.

```

protegido anulan OnBarUpdate vacio () {
    intervalo rangeInTicks = (intervalo) ((Alto [0] - Bajo [0]) / TickSize);
    myIntSeries. Juego (rangeInTicks);
}

```

IntSeries. Juego (intervalo *barsAgo*, bool *valor*) Usted también puede poner el valor para barras históricas por la inclusión de un valor "de *barsAgo*" que representa el número de hace unas barras que usted quiere que el valor de bool sea almacenado en.

Llamando el Reinicio () el método es único y puede ser muy potente para el desarrollo de indicador de encargo. Los objetos de IntSeries pueden sostener valores nulos que simplemente significa que usted no quiere almacenar un valor para la barra corriente. El reinicio () reinicializará el valor de índice corriente al nulo.

La comprobación de Valores Válidos es posible que usted pueda usar un objeto de IntSeries, pero decidir no poner un valor para una barra específica. Internamente, un valor falso realmente existe sin embargo, si usted quisiera comprobar para ver si era un valor válido que usted pone, usted puede comprobar el método siguiente.

IntSeries. ContainsValue (intervalo *barsAgo*) Vueltas un valor verdadero o falso.

La adquisición Valora Usted puede tener acceso a valores de objeto de IntSeries usando la sintaxis **IntSeries [intervalo *barsAgo*]** donde *barsAgo* representa el valor de datos n número de hace unas barras.

p379

www.ninjatrader.co

m

© NinjaTrader, LLC,
2005

```

protegido anulan OnBarUpdate vacio () {
    //Letras la Letra de valor de barra corriente y últimas ("Los valores son" +
    myIntSeries [0] + "" + myIntSeries [1]);
}

```

-O-

El Suceso Menos reciente (LRO)

La definición Devuelve el número de hace unas barras que la condición de prueba evaluada al verdadero dentro del período de mirada hacia atrás especificado expresada en barras. Esto comprueba de la barra lejana lejos y trabaja hacia la barra corriente. **Este método no trabajará en estrategias de multimarca de tiempo o multiinstrumento.**

Valor de retorno de método Un valor [internacional](#) que representa el número de hace unas barras. Devuelve un valor de -1 si la condición de prueba especificada no evaluara al verdadero dentro del período de mirada hacia atrás.

Sintaxis LRO (*Condición* [de condición](#), *caso* [internacional](#), *intervalo* *lookBackPeriod*)

La sintaxis para *la condición* debe ser escrita en la manera siguiente:

[el delegado](#) {[devuelve](#) su condición verdadera/falsa;}

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

condición	Un caso de expresión verdadero/falso	El suceso para comprobar (1 es el menos reciente, 2 es el 2do menos reciente etc...)
	lookBackPeriod	Número de barras para mirar hacia atrás para comprobar la condición de prueba

Ejemplos

```
//Imprime el alto precio del menos reciente barra sobre el 10 último intervalo de barras
barsAgo = LRO (delegado {vuelta Cerca [0]> Abierto [0];}, 1, 10); si (barsAgo>-1)
    La letra ("La barra alto era" + Alto [barsAgo]);
```

También ver [el Suceso más Reciente \(MRO\)](#)

-O-

Bajo

Definición Una colección de barra histórica precios bajos.

El Valor de propiedad `DataSet` escribe a máquina el objeto. Teniendo acceso a esta propiedad vía un valor de índice [\[el intervalo barsAgo\]](#) devuelve un doble valor que representa el precio de la barra referida.

p380

www.ninjatrader.com

Sintaxis Bajo [Bajo](#) [[intervalo](#) *barsAgo*]

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
//La barra corriente precio bajo dobla barLowPrice = Bajo [0];
//El precio bajo de hace 10 barras dobla barLowPrice = Bajo [10];
© NinjaTrader, LLC, 2005
```

```
//El valor de barra corriente de un 20 período valor medio móvil exponencial de  
precios bajos dobla el valor = Acuerdo Monetario Europeo (Bajo, 20) [0];
```

-o-

Mugidos

La definición Sostiene una serie de objetos [de DataSeries](#) sosteniendo la barra histórica precios bajos. Un objeto de DataSeries es añadido a esta serie llamando [Añadir \(\)](#) método en una Estrategia de Encargo. Su objetivo es proporcionar el acceso a los precios bajos de todos los objetos de Barras en la estrategia de multimarca de tiempo o un multiinstrumento.

Valor de propiedad Una serie de objetos de DataSeries.

Mugidos de sintaxis `[intervalo barSeriesIndex] [intervalo barsAgo]`

Propiedad De Estrategia de Encargo

Ejemplos

```
protegido anulan el vacío Inicializan () {  
    //Añade que unas Barras de 5 minutos se oponen a la estrategia y es  
    automáticamente  
    adjudicado  
    //un índice de objeto de Barras de 1 desde los datos primarios la estrategia  
    es dirigido  
    contra  
    //puesto por el UI toma el índice de 0. Añada (Instrumento, PeriodType. Minuto,  
    5);  
}  
  
protegido anulan OnBarUpdate vacío () {  
    //Compara el precio bajo de la barra primaria con el precio bajo de la barra  
    de 5 minutos si (Mugidos [0] [0] > Mugidos [1] [0])  
    La letra ("El precio bajo de la barra corriente es mayor");  
}
```

-o-

LowestBar ()

Definición

p381

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Devuelve el número de hace unas barras el valor de precios más bajo ocurrió para el período lookback. Si hay varios sucesos dentro del período lookback donde la barra era la barra más baja el suceso más reciente es devuelto.

El Valor de retorno de método Un valor [internacional](#) que representa el número de hace unas barras.

Sintaxis `LowestBar` (*serie* de `IDataSeries`, *intervalo* `lookBackPeriod`)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

serie Cualesquiera `DataSet` escriben a máquina el objeto, como un indicador, Cerca, Alto, Bajo etc. *lookBackPeriod* El número de barras para incluir en el cálculo

Ejemplos

```
//Imprime el número de hace unas barras para SMA más bajo (20) valor de la  
Letra de sesión corriente (lo Más bajo (SMA (20), Barras. BarsSinceSession  
- 1).ToString ());
```

-O-

MarketDataEventArgs

La definición Representa un cambio del nivel datos de mercado y es pasada como un parámetro en `OnMarketData ()` método.

Métodos y Propiedades

`MarketDataType` Los valores posibles son:

`MarketDataType`. Pregunte a `MarketDataType`. Ofrezca
`MarketDataType`. `DailyHigh` `MarketDataType`. `DailyLow`
`MarketDataType`. `DailyVolume` `MarketDataType`. Último
`MarketDataType`. `LastClose` (sesión previa cerca)

`Precio` Un **doble** valor

<<http://msdn2.microsoft.com/en-us/library/system.datetime.aspx>>

representación del Tiempo de precios Una estructura **de DateTime** que
representa el tiempo `ToString ()` Una representación **de cuerda** de

`MarketDataEventArgs` objeta el Volumen Un volumen de representación de
valor **internacional**

Ejemplos

```

protegido anulan OnMarketData vacío (MarketDataEventArgs e) {
    //Imprima algunos datos a la ventana Output si (e. MarketDataType ==
    MarketDataType. Último)
        Letra ("Último =" + e. Precio + "" + e. Volumen);
    más si (e. MarketDataType == MarketDataType. Pregunte)
        Letra ("Preguntan =" + e. Precio + "" + e. Volumen);
    más si (e. MarketDataType == MarketDataType. Oferta)
        Letra ("Oferta =" + e. Precio + "" + e. Volumen);}

```

Puntas 1. No todos los abastecedores de conectividad apoyan todo MarketDataTypes.

-O-

MarketDepthEventArgs

La definición Representa un cambio del nivel dos datos de mercado también conocidos como la profundidad de mercado y es pasada como un parámetro en [OnMarketDepth \(\)](#) método.

Métodos y Propiedades

MarketDataType Los valores posibles son:

MarketMaker MarketDataType. Pregunte a MarketDataType. Oferta
 Una **cadena** que representa el creador de mercado id Posición Un valor **internacional** que representa la posición basada cero en la escala de profundidad.
 Operación Representa la acción que usted debería tomar cuando la construcción de un nivel dos reserva.

Los valores posibles son:

Operación. Operación de encarte. Quite Operación. Actualización
 Precio Un **doble** valor <<http://msdn2.microsoft.com/en-us/library/system.datetime.aspx>>
representación del Tiempo **de precios** Una estructura **de DateTime** que representa el tiempo ToString () Una representación **de cadena** de MarketDataEventArgs objeto el Volumen Un volumen de representación de valor **internacional**

Ejemplos

```

protegido anulan OnMarketDepth vacío (MarketDepthEventArgs e) {
    //Imprima algunos datos a la ventana Output si (e. MarketDataType ==
    MarketDataType. Pregunte && e. Operación ==
    Operación. Actualización)
        La letra ("El más reciente piden cambiar es" + e. Precio + "" + e. Volumen);
}

```

-O-

El Suceso más Reciente (MRO)

Vueltas **de definición** el número de hace unas barras que la condición de prueba evaluada a verdadero dentro del especificado

el período de mirada hacia atrás expresado en barras. Esto comprueba de los trabajos de barra corrientes lejos (atrás) de ello. **Este método no trabajará en estrategias de multimarca de tiempo o multiinstrumento.**

Valor de retorno de método Un valor [internacional](#) que representa el número de hace unas barras. Devuelve un valor de -1 si la condición de prueba especificada no evaluara al verdadero dentro del período de mirada hacia atrás.

Sintaxis [MRO](#) (*Condición de condición, caso [internacional](#), intervalo [lookBackPeriod](#)*)

La sintaxis para *la condición* debe ser escrita en la manera siguiente:

[el delegado](#) {[devuelve](#) su condición verdadera/falsa;}

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

condición	Un caso de expresión verdadero/falso	El suceso para comprobar (1 es el más reciente, 2 es el 2do el más reciente etc...) lookBackPeriod
		Número de barras para mirar hacia atrás para comprobar la condición de prueba

Ejemplos

```
//Imprime el alto precio del más reciente barra sobre el 10 último intervalo de
barras barsAgo = MRO (delegado {vuelta Cerca [0]> Abierto [0];}, 1, 10); si
(barsAgo>-1)
    La letra ("La barra alto era" + Alto [barsAgo]);
```

También ver [el Suceso Menos reciente \(LRO\)](#)

-o-

Mediana

Definición Una colección de precios de mediana de barra históricos. Precio mediano = (Alto + Bajo) / 2.

El Valor de propiedad [DataSeries](#) escribe a máquina el objeto. Teniendo acceso a esta propiedad vía un valor de índice [\[el intervalo barsAgo\]](#) devuelve un doble valor que representa el precio de la barra referida.

[Mediana](#) de Mediana **de sintaxis** [\[intervalo barsAgo\]](#)

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos


```
//Precio de mediana de barra corriente
doble barMedianPrice = Mediana [0];

//El precio mediano de hace 10 barras dobla barMedianPrice = Mediana [10];

//El valor de barra corriente de un 20 período valor medio móvil exponencial de precios
medianos dobla el valor = Acuerdo Monetario Europeo (Mediana, 20) [0];
```

-0-

Medianas

La definición Sostiene una serie de objetos [de DataSeries](#) sosteniendo precios de mediana de barra históricos. Un objeto de DataSeries es añadido a esta serie llamando [Añadir \(\)](#) método en una Estrategia de Encargo. Su objetivo es proporcionar el acceso a los precios medianos de todos los objetos de Barras en la estrategia de multimarca de tiempo o un multiinstrumento.

Valor de propiedad Una serie de objetos de DataSeries.

Medianas de sintaxis [intervalo *barSeriesIndex*] [intervalo *barsAgo*]

Propiedad De Estrategia de Encargo

Ejemplos

```
protegido anulan el vacío Inicializan () {
    //Añade que unas Barras de 5 minutos se oponen a la estrategia y es
    automáticamente
    adjudicado
    //un índice de objeto de Barras de 1 desde los datos primarios la estrategia
    es dirigido
    contra
    //puesto por el UI toma el índice de 0. Añada (Instrumento, PeriodType. Minuto,
    5);
}

protegido anulan OnBarUpdate vacío () {
    //Compara el precio mediano de la barra primaria con la mediana de la barra de
    5 minutos
    precio
    si (Medianas [0] [0]> Medianas [1] [0])
        La letra ("El precio mediano de la barra corriente es mayor");
```

}

-o-

PercentComplete

La definición Consigue un valor que indica el por ciento de finalización de una barra.

- Siempre devolverá un valor del cero (0) trabajando con barras de variedad el • siempre devolverá un valor de uno (1) que representa el 100 % durante una estrategia backtest o dirigiendo una estrategia con [CalculateOnBarClose](#) puesto a un valor de • verdadero Por otra parte, nunca devolverá un valor de 1 ya que una barra completa no es conocida hasta que recibamos

p385

www.ninjatrader.com
© NinjaTrader, LLC, 2005

la 1ra marca de la siguiente barra así, usted también puede usar la propiedad [de FirstTickOfBar](#)

Valor de propiedad Un doble valor que representa un por ciento. p.ej el 50 % completo volvería 0.5.

Barras **de sintaxis**. PercentComplete

Propiedad De Indicador de Encargo, Estrategia de Encargo

La información de Acceso adicional pueden tener acceso a Esta propiedad sin una referencia nula se registra en OnBarUpdate () controlador de eventos. Cuando OnBarUpdate () el acontecimiento es provocado, siempre habrá un objeto de Barra que sostiene la propiedad de PercentComplete. Si usted desea tener acceso a esta propiedad en otra parte, compruebe la referencia nula primero. p.ej si (iBarras! = nulo)

-o-

OnBarUpdate ()

Definición OnBarUpdate () el método es llamado siempre que una barra sea actualizada. Si la propiedad ["CalculateOnBarClose"](#) es puesta al verdadero, sólo es llamado en el final de cada barra, por otra parte es llamado en cada marca de entrada. Esto es el método donde toda su estrategia o lógica de cálculo de corazón de indicador debería estar contenida.

- Para [mult-margen-de-tiempo y](#) estrategias [de instrumento](#), este método es llamado para cada objeto de Barras de una estrategia y usted **DEBEN** filtrar para los acontecimientos de actualización de barra exactos usando la propiedad ["BarsInProgress"](#) [contra la cual](#) usted quiere que su lógica de sistema ejecute.

El Valor de retorno de método Este método no devuelve un valor.

La sintaxis Ve el ejemplo abajo. El indicador NinjaScript y los magos de estrategia automáticamente generan la sintaxis de método para usted.

Método De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
protegido anulan OnBarUpdate vacío () {  
    si (CurrentBar <1)  
        vuelva;  
  
    //Compara la barra corriente con la barra previa si (Bajo [0]> Bajo [1])  
    La letra ("El precio bajo de la barra corriente es mayor");  
}
```

-0-

p386

www.ninjatrader.co

m

© NinjaTrader, LLC,

2005

OnMarketData ()

Definición

OnMarketData () método es llamado para cada cambio del nivel datos de mercado para el instrumento subyacente. Esto puede incluir, pero no es limitado con la oferta, pregunte, último precio y volumen.

- La programación a este método se considera la programación **avanzada** y expuesta para los programadores con experiencia • Esto son una corriente de datos de tiempo real y pueden ser la CPU intensiva si su código de programación es calculan intensivo (no óptimo) • Este método no es llamado en datos históricos (backtest)

El Valor de retorno de método Este método no devuelve un valor.

Parámetros de método [MarketDataEventArgs](#) e

Sintaxis Usted debe anular el método en su estrategia o indicador con la sintaxis siguiente.

```
protegido anulan OnMarketData vacío (MarketDataEventArgs e) {  
  
}
```

Método De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```

protegido anulan OnMarketData vacío (MarketDataEventArgs e) {
    //Imprima algunos datos a la ventana Output si (e. MarketDataType ==
    MarketDataType. Último)
        Letra ("Último =" + e. Precio + "" + e. Volumen);
    más si (e. MarketDataType == MarketDataType. Pregunte)
        Letra ("Preguntan =" + e. Precio + "" + e. Volumen);
    más si (e. MarketDataType == MarketDataType. Oferta)
        Letra ("Oferta =" + e. Precio + "" + e. Volumen);}

```

Las Muestras de Referencia adicionales código de referencia Adicional
 <<http://www.ninjatrader-support.com/vb/index.php>> muestras están disponibles
 NinjaScript sección de Recursos Educativa de nuestro [foro de apoyo](#).

Puntas 1. Con [multimarca de tiempo y estrategias de instrumento](#), OnMarketData será llamado para todos los instrumentos únicos en su estrategia. Use [BarsInProgress](#) para filtrar OnMarketData () método para un instrumento específico. 2. No abandone OnMarketData no usado () método declarado en su objeto de NinjaScript. Esto va innecesariamente ató una corriente de datos a su estrategia que usa ciclos de CPU innecesarios.

-0-

OnMarketDepth ()

p387

www.ninjatrader.com
 © NinjaTrader, LLC,

2005

Definición

OnMarketDepth () método es llamado para cada cambio del nivel dos datos de mercado (profundidad de mercado) para el instrumento subyacente. Este método puede ser usado para construir su propio nivel dos reservan.

- La programación a este método se considera la programación **avanzada** y expuesta para los programadores con experiencia • Esto son una corriente de datos de tiempo real y pueden ser la CPU intensiva si su código de programación es calculan intensivo (no óptimo) • Este método no es llamado en datos históricos (backtest)

El Valor de retorno de método Este método no devuelve un valor.

Parámetros de método [MarketDepthEventArgs](#) e

Sintaxis Usted debe anular el método en su estrategia o indicador con la sintaxis siguiente.

```

protegido anulan OnMarketDepth vacío (MarketDepthEventArgs e) {
}

```

Método De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```

protegido anulan OnMarketDepth vacío (MarketDepthEventArgs e) {
    //Imprima algunos datos a la ventana Output si (e. MarketDataType ==
    MarketDataType. Pregunte && e. Operación ==
    Operación. Actualización)
        La letra ("El más reciente piden cambiar es" + e. Precio + " " + e. Volumen); }

```

Las Muestras de Referencia adicionales código de referencia Adicional
<<http://www.ninjatrader-support.com/vb/index.php>> muestras están disponibles
 NinjaScript sección de Recursos Educativa de nuestro [foro de apoyo](#).

Puntas 1. Con [multimarca de tiempo y estrategias de instrumento](#), OnMarketDepth será llamado para todos los instrumentos únicos en su estrategia. Use [BarsInProgress](#) para filtrar OnMarketDepth () método para un instrumento específico. 2. No abandone OnMarketDepth no usado () método declarado en su objeto de NinjaScript. Esto va innecesariamente ató una corriente de datos a su estrategia que usa ciclos de CPU innecesarios.

-o-

Abierto

Definición Una colección de barra histórica precios iniciales.

El Valor de propiedad [DataSeries](#) escribe a máquina el objeto. Teniendo acceso a esta propiedad vía un valor de índice [\[el intervalo barsAgo\]](#) devuelve a

p388

www.ninjatrader.com
 © NinjaTrader, LLC, 2005

doble el valor que representa el precio de la barra referida.

Sintaxis Abierta [Abierto \[intervalo barsAgo\]](#)

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```

//La barra corriente precio inicial dobla barOpenPrice = Abierto [0];
//El precio inicial de hace 10 barras dobla barOpenPrice = Abierto [10];
//El valor de barra corriente de un 20 período valor medio móvil simple de precios
iniciales dobla el valor = SMA (Abierto, 20) [0];

```

-o-

Se abre

La definición Sostiene una serie de objetos [de DataSeries](#) sosteniendo la barra histórica precios abiertos. Un objeto de DataSeries es añadido a esta serie llamando [Añadir \(\)](#) método en una Estrategia de Encargo. Su objetivo es proporcionar el acceso a los precios abiertos de todos los objetos de Barras en la estrategia de multimarca de tiempo o un multiinstrumento.

Valor de propiedad Una serie de objetos de DataSeries.

La sintaxis Abre [el intervalo *barSeriesIndex*] [intervalo *barsAgo*]

Propiedad De Estrategia de Encargo

Ejemplos

```
protegido anulan el vacío Inicializan () {
    //Añade que unas Barras de 5 minutos se oponen a la estrategia y es
    automáticamente
    adjudicado
    //un índice de objeto de Barras de 1 desde los datos primarios la estrategia
    es dirigido
    contra
    //puesto por el UI toma el índice de 0. Añada (Instrumento, PeriodType. Minuto,
    5);
}

protegido anulan OnBarUpdate vacío () {
    //Compara el precio abierto de la barra primaria con el precio abierto de la
    barra de 5 minutos si (Abierto [0] [0]> Se abre [1] [0])
        La letra ("El precio abierto de la barra corriente es mayor");
}
```

-0-

p389

www.ninjatrader.com
© NinjaTrader, LLC,

2005

Período

Definición

Las Barras objetan el tipo de período y el intervalo.

Valor de propiedad Un objeto de Período.

Barras **de sintaxis**. Período. Barras de Id. Período. Valor

* Id puede ser PeriodType de valor. Marca, PeriodType. Volumen, PeriodType. En segundo lugar, PeriodType. Variedad, PeriodType. Minuto, PeriodType. Día, PeriodType. Semana, PeriodType. Mes, PeriodType. Año * Valor devolverá un valor entero

NOTE: aconsejamos usar la propiedad **de BarsPeriod** para una Estrategia.

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
//Sólo cuente si hay una 100 carta de marca o mayor protegido anulan OnBarUpdate
vacío () {
```

```

si (Barras. Período. Id == PeriodType. Marca && Barras. Período. Valor> = 100)
{
    //Lógica de cálculo de indicador aquí
}

```

La información de Acceso adicional pueden tener acceso a Esta propiedad sin una referencia nula se registra en OnBarUpdate () controlador de eventos. Cuando OnBarUpdate () el acontecimiento es provocado, siempre habrá un objeto de Barra que sostiene la propiedad de Período. Si usted desea tener acceso a esta propiedad en otra parte, compruebe la referencia nula primero. p.ej si (iBarras! = nulo)

-0-

Rebelión ()

Los Controles **de definición** de una condición creciente que es verdad cuando el valor corriente es mayor que el valor de hace 1 barra.

El Valor de retorno de método Este método vueltas **verdaderas** si una condición creciente está presente; por otra parte, **falso**.

Rebelión de sintaxis (*serie de IDataSeries*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

serie Cualesquiera DataSet escriben a máquina el objeto, como un indicador, Cerca, Alto, Bajo etc.

p390

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Ejemplos

```

//Si el 20 período SMA se eleva (en uptrend) va mucho tiempo si (Elevándose (SMA
(20))
EnterLong ();

```

-0-

SessionBreak

La definición Consigue un valor que indica si la llamada a OnBarUpdate () método es la primera llamada en una nueva sesión.

La propiedad Valora Este propiedad vueltas **verdaderas** si la llamada a OnBarUpdate () es la primera llamada de la sesión; por otra parte, **falso**.

Barras **de sintaxis**. SessionBreak

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
//El método de OnBarUpdate protegido anula OnBarUpdate vacío () {  
    si (Barras. SessionBreak)  
        //Reinicio o tienda algunos valores aquí  
}
```

-0-

Cuesta ()

La definición La cuesta es una medida del escarpado de una línea. Esto es el cálculo de subida sobre el dirigido y no ser confundido con la cuesta de regresión lineal.

El Valor de retorno de método Este método devuelve un **dobles** valor que indica la cuesta de una línea

Cuesta de sintaxis (*serie de IDataSeries, intervalo startBarsAgo, intervalo endBarsAgo*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

serie Cualesquiera DataSeries escriben a máquina el objeto, como un indicador, Cerca, Alto, Bajo etc. startBarsAgo Especifica el punto de partida de una línea endBar Especifica que el punto que se termina de alineación

Ejemplos

```
//Imprime la cuesta del 20 periodo valor medio móvil simple de la 10 última Letra  
de barras (Cuesta (SMA (20), 9, 0));
```

p391

www.ninjatrader.com
© NinjaTrader, LLC,

2005

-0-

Clase de StringSeries

Definición

StringSeries es un tipo especial de la estructura de datos que sostiene una serie de valores de cuerda y siempre contiene el mismo número de elementos que barras en una carta. Ver la Clase [de DataSeries](#) para la información relacionada.

Creación de Objetos de StringSeries De crear un objeto de StringSeries:

1. Defina una variable ("myStringSeries" usado en este ejemplo) del tipo StringSeries que creará que StringSeries se opone 2. En Inicializar () método, cree un nuevo objeto de

StringSeries y adjudíquelo a la variable "myStringSeries"

```
#region Variables
    StringSeries privado myStringSeries; //Definen una variable de StringSeries
#endregion

//Cree un objeto de StringSeries y adjudique esto a la variable protegida anula el
vacío Inicializan () {
    myStringSeries = nuevo StringSeries (esto); // "esto" se refiere al
    indicador, o estrategia
    Objeto de StringSeries
    //sí mismo. Esto sincroniza el
    //a barras de datos históricas}
```

* los objetos de StringSeries pueden ser usados en la serie suplementaria en la estrategia de instrumento y una multimarca de tiempo. Por favor ver nuestro foro de apoyo sección de muestras de referencia de NinjaScript para más información.

El ajuste de Valores - StringSeries. Juego () & StringSeries. Reinicio () Usted puede ponerse valor (enchufe un valor) en un objeto de StringSeries llamando el Juego () método.

StringSeries. El juego (valor de cuerda) Ajuste de un valor en un objeto de StringSeries es automáticamente alineado a la barra corriente evaluada. Esto asegura que los valores a través de todos los objetos de StringSeries siempre están en la sincronización por el índice [de CurrentBar](#). Las muestras de código siguientes demuestran el cálculo de la variedad de cada barra y almacenaje del valor en un objeto de StringSeries.

```
protegido anulan OnBarUpdate vacío () {
    //Sólo almacene algún texto myStringSeries. Juego (CurrentBar. ToString ());
}
```

StringSeries. Juego (intervalo barsAgo, ensarte el valor) Usted también puede poner el valor para barras históricas por la inclusión de un valor "de barsAgo" que representa el número de hace unas barras que usted quiere que el valor de cuerda sea almacenado en.

Llamando el Reinicio () el método es único y puede ser muy potente para el desarrollo de indicador de encargo. Los objetos de StringSeries pueden sostener valores nulos que simplemente significa que usted no quiere almacenar un valor para la barra corriente. El reinicio () reinicializará el valor de índice corriente al nulo.

La comprobación de Valores Válidos

Es posible que usted pueda usar un objeto de StringSeries, pero decidir no poner un valor para una barra específica. Internamente, un valor falso realmente existe sin embargo, si usted quisiera comprobar para ver si era un valor válido que usted pone, usted puede comprobar el método siguiente.

StringSeries.ContainsValue (intervalo barsAgo) Vuelta un valor verdadero o falso.

La adquisición Valora Usted puede tener acceso a valores de objeto de StringSeries usando la sintaxis **StringSeries [intervalo barsAgo]** donde barsAgo representa el valor de datos n número de hace unas barras.

```
protegido anulan OnBarUpdate vacío () {  
    //Letras la Letra de valor de barra corriente y últimas ("Los valores son" +  
    myStringSeries [0] + "" + myStringSeries [1]);  
}
```

-o-

TickCount

Definición El número total de marcas de la barra corriente.

La propiedad Valora un valor **internacional** que representa el número total de marcas de la barra corriente.

Barras **de sintaxis**. TickCount

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
//Letras la cuenta de marca a la ventana Print de salida ("El cargo de marca por  
la barra corriente es" + Barras. TickCount. ToString ());
```

La información de Acceso adicional pueden tener acceso a Esta propiedad sin una referencia nula se registra en OnBarUpdate () controlador de eventos. Cuando OnBarUpdate () el acontecimiento es provocado, siempre habrá un objeto de Barra que sostiene la propiedad de TickCount. Si usted desea tener acceso a esta propiedad en otra parte, compruebe la referencia nula primero. p.ej si (iBarras! = nulo)

-o-

TickSize

Definición El valor de fluctuación mínimo; el valor de 1 marca para el instrumento correspondiente.

NOTE: no deberían tener acceso a esta propiedad dentro de [Inicializar \(\)](#) método.

La propiedad Valora un doble valor que representa la fluctuación mínima de un instrumento.

Sintaxis

TickSize

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
//Letras el ticksize a la ventana Print de salida ("El ticksize de este instrumento es" + TickSize. ToString ());  
  
//Imprime el valor de la barra corriente bajo menos una talla de marca doble valor  
= Bajo [0] - TickSize; Letra (valor. ToString ());
```

-o-

Tiempo

Definición Una colección de valores de sello de tiempo de barra históricos.

El Valor de propiedad TimeSeries escribe a máquina el objeto.

El Tiempo de Tiempo **de sintaxis** [intervalo barsAgo] (devuelve una estructura [de DateTime](#))

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
//Imprime la Letra de sello de tiempo de barra corriente (Tiempo [0].ToString ());  
  
//Los controles si el tiempo corriente es mayor que el sello de tiempo de barra si  
(DateTime. Ahora. Marcas> Tiempo [0].Ticks)  
La letra ("Hacen algo");
```

-o-

Tiempos

La definición Sostiene una serie de objetos de TimeSeries sosteniendo tiempos de barra históricos. Un objeto de TimeSeries es añadido a esta serie llamando [Añadir \(\)](#) método en una Estrategia de Encargo. Su objetivo es proporcionar el acceso a los tiempos de todos los objetos de Barras en la estrategia de multimarca de tiempo o un multiinstrumento.

Valor de propiedad Una serie de objetos de TimeSeries.

Tiempos de sintaxis [intervalo barSeriesIndex] [intervalo barsAgo]

Propiedad De Estrategia de Encargo

Ejemplos

```

protegido anulan el vacío Inicializan () {
    //Añade que unas Barras de 5 minutos se oponen a la estrategia y es
    automáticamente
    adjudicado
    //un índice de objeto de Barras de 1 desde los datos primarios la estrategia
    es dirigido
    contra
    //puesto por el UI toma el índice de 0. Añada (Instrumento, PeriodType. Minuto,
    5);
}

protegido anulan OnBarUpdate vacío () {
    //Compara el tiempo de la barra primaria con el tiempo de la barra de 5 minutos
    si (Tiempos [0] [0].Ticks> Tiempos [1] [0].Ticks)
        La letra ("El tiempo de la barra corriente es mayor");
}

```

-0-

ToDay ()

La definición Calcula un día de representación de valor entero. NinjaScript usa las estructuras [de DateTime.NET](#)

<<http://msdn2.microsoft.com/en-us/library/system.datetime.aspx>>which puede ser complicado para programadores de principiante. Si usted es familiar con C# usted puede usar directamente propiedades de estructura de DateTime y métodos para la fecha y las comparaciones de tiempo por otra parte usan este método y [ToTime \(\)](#) método.

La representación de número entero del día está en el formato siguiente yyyyMMdd donde el 8 de enero de 2007 sería 20070108.

El Valor de retorno de método Un valor internacional que representa la fecha.

Sintaxis ToDay (*tiempo* de DateTime)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

tiempo Una estructura de DateTime, como el Tiempo [0] - Ve la propiedad [de Tiempo](#)

Ejemplos

```

//Compare la fecha de la barra corriente al 15 de septiembre de
2006 si (ToDay (Tiempo [0])> 20060915)
    //Haga algo

```

-0-

TotalTicks

La definición Consigue un valor para el total cummulative de todas las marcas para todas las sesiones.

Valor de propiedad Un número total de representación de valor internacional de marcas.

p395

www.ninjatrader.com

Barras **de sintaxis**. TotalTicks

Propiedad De Indicador de Encargo, Estrategia de Encargo

Información de Acceso adicional

© NinjaTrader, LLC, 2005

Pueden tener acceso a esta propiedad sin una referencia nula se registran en OnBarUpdate () controlador de eventos. Cuando OnBarUpdate () el acontecimiento es provocado, siempre habrá un objeto de Barra que sostiene la propiedad de TotalTicks. Si usted desea tener acceso a esta propiedad en otra parte, compruebe la referencia nula primero. p.ej si (iBarras! = nulo)

-o-

TriggerCustomEvent ()

La definición Este método proporciona una manera de usar sus propios acontecimientos de encargo (como un objeto de Temporizador) de modo que los índices de NinjaScript internos y las agujas sean correctamente puestos antes del procesamiento del código de usuario provocado por su acontecimiento de encargo. Llamando este acontecimiento, NinjaTrader sincronizará todas las agujas internas y luego llamará su controlador de eventos de encargo donde su código de usuario está localizado.

El Valor de retorno de método Este método no tiene un valor de retorno.

Sintaxis TriggerCustomEvent (CustomEvent customEvent, *objeto el estado*)
TriggerCustomEvent (CustomEvent customEvent, *intervalo barsIndex, estado de objeto*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

CustomEvent Delegado de su método de acontecimiento de encargo barsIndex

El índice de la *serie de barra* usted quiere sincronizar para declarar

Cualquier objeto que usted quiera pasó en su método de acontecimiento de encargo

Ejemplos

```
//Su objeto de temporizador señala el controlador de eventos OnTimerTick vacío
privado () {
    //No trate su código aquí, pero en cambio llame TriggerCustomEvent ()//método
```

```

        y trate su código en el método de MyCustomHandler TriggerCustomEvent (nuevo
        CustomEvent (MyCustomHandler, 0, "myText"));
    }

    MyCustomHandler vacío privado (objetan el estado) {
        La letra ((cuerda) declaran + "" + CurrentBar. ToString ());
    }

```

-o-

ToTime ()

La definición Calcula un tiempo de representación de valor entero. NinjaScript usa la estructura de DateTime.NET

p396

www.ninjatrader.com
© NinjaTrader, LLC, 2005

que puede ser complicado para programadores de principiante. Si usted es familiar con C# usted puede usar directamente propiedades de estructura de DateTime y métodos para la fecha y las comparaciones de tiempo por otra parte usan este método y [ToDay \(\)](#) método.

La representación de número entero del tiempo está en el formato siguiente Hmms donde 7:30 serían 73000 y 14:15:12 serían 141512.

El Valor de retorno de método Un valor [internacional](#) que representa el tiempo.

Sintaxis [ToTime](#) (*tiempo de DateTime*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

tiempo Una estructura de DateTime, como el Tiempo [0] - Ve la propiedad [de Tiempo](#)

Ejemplos

```

//Sólo comercio entre las 7:45 y 13:45 si (ToTime (Tiempo [0])> = 74500 && ToTime
(Tiempo [0]) <= 134500) {
    //La lógica de estrategia va aquí
}

```

-o-

Típico

Definición Una colección de barra histórica precios típicos. Precio típico = (Alto + Bajo + Cerca) / 3.

El Valor de propiedad [DataSeries](#) escribe a máquina el objeto. Teniendo acceso a esta propiedad vía un valor de índice [\[el intervalo barsAgo\]](#) devuelve un doble valor que

representa el precio de la barra referida.

Sintaxis `Típica` `Típico` [`intervalo` `barsAgo`]

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
//La barra corriente precio típico dobla barTypicalPrice = Típico [0];  
//El precio mediano de hace 10 barras dobla barTypicalPrice = Típico [10];  
//El valor de barra corriente de un 20 período valor medio móvil exponencial de precios  
típicos dobla el valor = Acuerdo Monetario Europeo (Típico, 20) [0];
```

-0-

p397

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Typicals

Definición

Sostiene una serie de objetos [de DataSeries](#) sosteniendo la barra histórica precios típicos. Un objeto de DataSeries es añadido a esta serie llamando [Añadir \(\)](#) método en una Estrategia de Encargo. Su objetivo es proporcionar el acceso a los precios típicos de todos los objetos de Barras en la estrategia de multimarca de tiempo o un multiinstrumento.

Valor de propiedad Una serie de objetos de DataSeries.

Sintaxis `Typicals` [`intervalo` `barSeriesIndex`] [`intervalo` `barsAgo`]

Propiedad De Estrategia de Encargo

Ejemplos

```
protegido anulan el vacío Inicializan () {  
    //Añade que unas Barras de 5 minutos se oponen a la estrategia y es  
    automáticamente  
    adjudicado  
    //un índice de objeto de Barras de 1 desde los datos primarios la estrategia  
    es dirigido  
    contra  
    //puesto por el UI toma el índice de 0. Añada (Instrumento, PeriodType. Minuto,  
    5);  
}  
  
protegido anulan OnBarUpdate vacío () {  
    //Compara el precio típico de la barra primaria con la barra de 5 minutos típica  
    precio  
    si (Typicals [0] [0]> Typicals [1] [0])  
        La letra ("El precio típico de la barra corriente es mayor");  
}
```

-0-

Volumen

Definición Una colección de valores de volumen de barra históricos.

El Valor de propiedad VolumeSeries escribe a máquina el objeto. Teniendo acceso a esta propiedad vía un valor de índice [\[el intervalo barsAgo\]](#) devuelve un [doble](#) valor que representa el volumen de la barra referida.

[Volumen](#) de Volumen **de sintaxis** [\[intervalo barsAgo\]](#)

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

p398

www.ninjatrader.com
© NinjaTrader, LLC, 2005

```
//Método de OnBarUpdate
protegido anulan OnBarUpdate vacío () {
    //Es el volumen corriente mayor que dos veces el volumen de la barra previa si
    (el Tomo [0]> el Tomo [1] * 2)
        Letra ("Hemos aumentado el volumen");

    //Es el volumen corriente mayor que el 20 valor medio móvil de período de
    volumen
    si (el Tomo [0]> SMA (Volumen, 20) [0])
        Letra ("Aumentando volumen");
}
```

-0-

Volúmenes

La definición Sostiene una serie de objetos de VolumeSeries sosteniendo volúmenes de barra históricos. Un objeto de DataSeries es añadido a esta serie llamando [Añadir \(\)](#) método en una Estrategia de Encargo. Su objetivo es proporcionar el acceso a los volúmenes de todos los objetos de Barras en la estrategia de multimarca de tiempo o un multiinstrumento.

Valor de propiedad Una serie de objetos de VolumeSeries.

[Volúmenes de sintaxis](#) [\[intervalo barSeriesIndex\]](#) [\[intervalo barsAgo\]](#)

Propiedad De Estrategia de Encargo

Ejemplos


```

protegido anulan el vacío Inicializan () {
    //Añade que unas Barras de 5 minutos se oponen a la estrategia y es
    automáticamente
    adjudicado
    //un índice de objeto de Barras de 1 desde los datos primarios la estrategia
    es dirigido
    contra
    //puesto por el UI toma el índice de 0. Añada (Instrumento, PeriodType. Minuto,
    5);
}

protegido anulan OnBarUpdate vacío () {
    //Compara el volumen de la barra primaria con el volumen de la barra de 5 minutos
    si (Volúmenes [0] [0]> Volúmenes [1] [0])
        La letra ("El precio de volumen de la barra corriente es mayor");
}

```

-o-

Compartido Dibujan Métodos y Propiedades

Descripción

Usted puede usar NinjaScript para dibujar formas de encargo, líneas, texto y colores en su precio y paneles de indicador.

p399

www.ninjatrader.com
© NinjaTrader, LLC,

2005

Puntas

1. La gráfica de encargo para indicadores de encargo puede ser pintada en el panel de precios o en indicador panel. Usted podría hacer mostrar por ejemplo un indicador de encargo en un panel de indicador aún han asociado la gráfica de encargo pintada en el panel de precios. La propiedad "DrawOnPricePanel" es puesta al verdadero como la falta el que significa que la gráfica de encargo siempre será pintada en el panel de precios aun si el indicador es trazado en un panel separado. Si usted quiere que su gráfica de encargo sea trazada en el panel de indicador, ponga esta propiedad al falso en Inicializar () el método de su indicador de encargo.
2. Póngase los valores de etiqueta únicos para cada uno dibujan el objeto. Una broma común debe incorporar la barra número como parte del identificador de etiqueta único. Por ejemplo, si usted quiso dibujar un punto que indicó una condición de compra encima de una barra, usted podría expresarlo:
DrawDot (CurrentBar. ToString () + "Compran", 0, Alto [0] + TickSize, Color. ForestGreen).
3. Dibuje métodos no trabajará si los llaman de Inicializar () método.

BackColor

La definición Elige el panel de carta atrás color para la barra corriente.

La propiedad Valora una estructura [en color](#) que representa el color de esta barra de precios.

Sintaxis BackColor

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
//Elige el panel de carta atrás color a BackColor verde pálido = Color. PaleGreen;  
  
//Elige el color trasero al juego de color de falta en la ventana BackColor de diálogo  
de propiedades de carta = Color. Vacío;  
  
//¿Elige el color trasero al granate cuándo la cotización de cierre está menos que  
el 20 período SMA//y al verde lima cuando encima (ver la imagen abajo) BackColor  
= SMA (20) [0]> = Cerca [0]? Color. Granate: Color. LimeGreen;
```

-0-

BarColor

La definición Elige el color de una barra de precios.

Valor de propiedad

Una estructura [en color](#) que representa el color de esta barra de precios.

Sintaxis BarColor

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
//Elige el color de barra a BarColor amarillo = Color. Amarillo;  
  
//Elige el color de barra a su color de falta como definido en el diálogo de propiedades  
de carta BarColor = Color. Vacío;  
  
//Elige el color de barra al amarillo si 20 SMA están encima de 50 SMA y el cierre//el  
precio está encima de 20 SMA (ver la imagen abajo) si (SMA (20) [0]> SMA (50) [0]  
&& Cerca [0]> SMA (20) [0])  
    BarColor = Color. Amarillo;
```

-0-

CandleOutlineColor

La definición Elige el color de contorno de un palo de vela.

p401

www.ninjatrader.com
© NinjaTrader, LLC, 2005

* Si usted pone la propiedad [de BarColor](#) y la propiedad de CandleOutlineColor no es puesto, el color de contorno tomará el color elegido por la propiedad de BarColor.

La propiedad Valora una estructura [en color](#) que representa el color de esta barra de precios.

Sintaxis CandleOutlineColor

Propiedad De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
//Elige el color de contorno de vela a CandleOutlineColor negro = Color. Negro;
```

-o-

Dibujo de encargo

No hay prácticamente ningunos límites para la costumbre que usa un panel de carta. Ya que NinjaScript está basado en Microsoft C#, el Sistema entero. El dibujo namespace está disponible para el uso. El dibujo de encargo es un tema de programación avanzado y sólo debería ser considerado por programadores familiares con Microsoft C# y el Marco.NET.

Por favor ver los indicadores siguientes para muestras de referencia del dibujo de encargo anulando el Complot () el método de la clase base de Indicador:

- CustomPlotSample
- TickCounter
- VolumeZones

-o-

DrawAndrewsPitchfork ()

La definición Dibuja la Horca de Andrew.

DrawAndrewsPitchfork de sintaxis (etiqueta de cuerda, bool autoescala, intervalo anchor1BarsAgo, dobles anchor1Y, intervalo anchor2BarsAgo, doblan anchor2Y, intervalo anchor3BarsAgo, doblan anchor3Y)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta	Un usuario definió id único usado para referirse al objeto de empate. Por ejemplo,		
	si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibujo el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez.		
autoescala	Determina si el objeto de empate será incluido en anchor1BarsAgo de escala		
de eje Y	El número de hace unas barras (x valor) del 1er punto de ancla anchor1Y		
	El valor y del 1er punto de ancla anchor2BarsAgo	El número de hace unas barras (x valor) del 2do punto de ancla	

p402

www.ninjatrader.com
© NinjaTrader, LLC, 2005

anchor2Y El valor y del 2do punto de ancla anchor3BarsAgo El

número de hace unas barras (x valor) del 3er punto de ancla anchor3Y El
valor y del 3er punto de ancla

Ejemplos

```
DrawAndrewsPitchfork ("tag1", verdadero, 4, Bajo [4], 3, Alto [3], 1, Bajo  
[1]);
```

-o-

DrawArrowDown ()

La definición Dibuja una flecha que señala abajo.

DrawArrowDown de sintaxis (*etiqueta de cuerda, bool autoescala, intervalo
barsAgo, doble y, color en Color*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta Un usuario definió id único usado para referirse al objeto de empate. Por
ejemplo, si usted pasa en un valor de "myTag", cada vez que esta etiqueta es
usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son
usadas cada vez, un nuevo objeto de empate será creado cada vez.
autoescala Determina si el objeto de empate será incluido en barsAgo de escala de eje Y
La barra el objeto será dibujada en. Un valor de 10 sería hace 10 barras. y Los y
valoran el color El color de objeto de empate ([referencia](#))

Ejemplos

```
//Pinta un rojo abajo flecha en la barra corriente 1 marca encima de DrawArrowDown  
alto ("tag1", verdadero, 0, Alto [0] + TickSize, Color. Rojo);  
  
//Pinta un azul abajo arrown en un tres modelo de inversión de barra (ver la imagen  
abajo) si (Alto [2]> Alto [3] && Alto [1]> Alto [2] && Cerca [0] <Abierto [0])  
DrawArrowDown (CurrentBar. ToString (), verdadero, 0, Alto [0] + TickSize,  
Color. Azul);
```

-0

DrawArrowLine ()

La definición Dibuja una línea de flecha.

Sintaxis

DrawArrowLine (*etiqueta de cuerda, intervalo startBarsAgo, doble startY, intervalo endBarsAgo, doble endY, color en Color*) **DrawArrowLine** (*etiqueta de cuerda, bool autoescala, intervalo startBarsAgo, dobles startY, intervalo endBarsAgo, doblan endY, color en Color, DashStyle dashStyle, anchura internacional*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta ejemplo,	Un usuario definió id único usado para referirse al objeto de empate. Por si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez.
autoescala	Determina si el objeto de empate será incluido en la escala de eje Y. Falta

el valor es [falso](#). startBarsAgo La barra inicial (coordenada de eje X) donde el objeto de empate será dibujado. Por ejemplo, un valor de 10 pintaría el empate objetan 10 barras atrás.

startY El comienzo y valora la coordenada donde el objeto de empate será dibujado

endBarsAgo La barra de final (coordenada de eje X) donde el objeto de empate terminará

endY El final y valora la coordenada donde el objeto de empate terminará el color El color de objeto de empate ([referencia](#))

p404

www.ninjatrader.co
m

dashStyle

DashStyle. Carrera
© NinjaTrader, LLC, 2005

DashStyle. DashDot
DashStyle. DashDotDot
DashStyle. Dot
DashStyle. Sólido
([referencia](#))

anchura La anchura del objeto de empate

Ejemplos

```
//Dibuja una línea de flecha verde lima punteada DrawArrowLine  
("tag1", falso, 10, 1000, 0, 1001, Color. LimeGreen, DashStyle.  
Punto, 2);
```

-o-

DrawArrowUp ()

La definición Dibuja una flecha que señala.

DrawArrowUp de sintaxis (*etiqueta de cuerda*, *bool autoescala*, *intervalo barsAgo*, *doble y*, *color en Color*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta ejemplo, Un usuario definió id único usado para referirse al objeto de empate. Por si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez.

autoescala Determina si el objeto de empate será incluido en barsAgo de escala de eje Y La barra el objeto será dibujada en. Un valor de 10 sería hace 10 barras. y Los y valoran el color El color de objeto de empate ([referencia](#))

Ejemplos

```
//Pinta un rojo abajo flecha en la barra corriente 1 marca debajo de
DrawArrowUp bajo ("tag1", verdadero, 0, Bajo [0] - TickSize, Color.
Rojo);
```

-0-

DrawDiamond ()

La definición Dibuja un diamante.

DrawDiamond de sintaxis (*etiqueta* [de cuerda](#), [bool autoescala](#), [intervalo barsAgo](#), [doble y](#), [color en Color](#))

Método De

p405

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Indicador de encargo, Estrategia de Encargo

Parámetro

s
etiqueta

autoescale barsAgo ycolor

Ejemplos

Un usuario definió id único usado para referirse al objeto de empate. Por ejemplo, si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez. Determina si el objeto de empate será incluido en la escala de eje Y La barra el objeto será dibujado en. Un valor de 10 sería hace 10 barras. Los y valoran El color de objeto de empate ([referencia](#))

```
//Pinta un diamante rojo en la barra corriente 1 marca debajo de
DrawDiamond bajo ("tag1", verdadero, 0, Bajo [0] - TickSize, Color.
```


Rojó);

-0-

DrawDot ()

La definición Dibuja un punto.

DrawDot de sintaxis (*etiqueta de cuerda, bool autoescala, intervalo barsAgo, doble y, color en Color*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta

autoescale barsAgo ycolor

Ejemplos

Un usuario definió id único usado para referirse al objeto de empate. Por ejemplo, si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez. Determina si el objeto de empate será incluido en la escala de eje Y La barra el objeto será dibujado en. Un valor de 10 sería hace 10 barras. Los y valoran El color de objeto de empate ([referencia](#))

```
//Pinta un punto rojo en la barra corriente 1 marca debajo de
DrawDot bajo ("tag1", verdadero, 0, Bajo [0] - TickSize, Color.
Rojó);
```

-0-

DrawEllipse ()

La definición Dibuja una elipse.

p406

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Sintaxis

DrawEllipse (*etiqueta de cuerda, intervalo startBarsAgo, doble startY, intervalo endBarsAgo, doble endY, color en Color*) **DrawEllipse** (*etiqueta de cuerda, bool autoescala, intervalo startBarsAgo, dobles startY, intervalo endBarsAgo, doblan endY, color en Color, Color backColor, opacidad internacional*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta ejemplo,	Un usuario definió id único usado para referirse al objeto de empate. Por si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibujo el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez.
autoescala	Determina si el objeto de empate será incluido en la escala de eje Y. Falta el valor es falso .
startBarsAgo	La barra inicial (coordenada de eje X) donde el objeto de empate será dibujado. Por ejemplo, un valor de 10 pintaría el empate objetan 10 barras atrás.
startY	El comienzo y valora la coordenada donde el objeto de empate será dibujado
endBarsAgo	La barra de final (coordenada de eje X) donde el objeto de empate terminará
endY	El final y valora la coordenada donde
	<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfsystemdrawingcolormemberstopic.asp> el objeto de empate terminará el color
	El color de objeto de empate (referencia) backColor El color llenar para el objeto (referencia) opacidad Pone el nivel de transparencia para el color llenar. Valores válidos entre 1 -
	10.

Ejemplos

```
//Dibuja una elipse azul de las 10 barras bajas atrás al nivel más alto de 5 barras atrás
DrawEllipse ("tag1", 10, Bajo [10] - TickSize, 5, Alto [5] + TickSize, Color. Azul);
```

```
//Dibuja una elipse azul de las 10 barras bajas atrás al nivel más alto de 5 barras atrás con//un llenar en color o verde pálido con un nivel de transparencia de 2
DrawEllipse ("tag1", falso, 10, Bajo [10] - TickSize, 5, Alto [5] + TickSize, Color. PaleGreen, Color. PaleGreen, 2);
```

-0-

DrawExtendedLine ()

La definición Dibuja una línea con puntos finales infinitos.

DrawExtendedLine de sintaxis (*etiqueta de cuerda*, *intervalo* *startBarsAgo*, *doble* *startY*, *intervalo* *endBarsAgo*, *doble* *endY*, *color* en *Color*) **DrawExtendedLine** (*etiqueta de cuerda*, *bool* *autoescala*, *intervalo* *startBarsAgo*, *dobles* *startY*, *intervalo* *endBarsAgo*, *doblan* *endY*, *color* en *Color*, *DashStyle* *dashStyle*, *anchura* *internacional*)

Método De Indicador de Encargo, Estrategia de Encargo

etiqueta

autoescala

startBarsAgo

los startY endBarsAgo endY colorean dashStyle

anchura

Ejemplos

© NinjaTrader, LLC, 2005

Los parámetros Un usuario definió id único solían referirse al objeto de empate. Por ejemplo, si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez. Determina si el objeto de empate será incluido en la escala de eje Y. El valor por defecto es **falso**. La barra inicial (coordenada de eje X) donde el objeto de empate será dibujado. Por ejemplo, un valor de 10 pintaría el empate objetan 10 barras atrás. El comienzo y valora la coordenada donde el objeto de empate será dibujado La barra de final (coordenada de eje X) donde el objeto de empate terminará El final y coordenada de valor **donde**

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfssystemdrawingcolormemberstopic.asp> el objeto de empate terminará El color de objeto de empate [\(referencia\)](#)

DashStyle. Carrera

DashStyle. DashDot DashStyle. DashDotDot DashStyle. Dot DashStyle. Sólido [\(referencia\)](#)

La anchura del objeto de empate

```
//Atrae DrawExtendedLine verde lima punteado ("tag1", falso, 10, 1000, 0, 1001, Color. LimeGreen, DashStyle. Punto, 2);
```

-o-

DrawFibonacciCircle ()

La definición Dibuja el círculo fibonacci.

DrawFibonacciCircle de sintaxis (*etiqueta de cuerda*, *bool autoescala*, *intervalo*

startBarsAgo, dobles startY, intervalo endBarsAgo, doblan endY)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta

autoescala startBarsAgo

startY

Un usuario definió id único usado para referirse al objeto de empate. Por ejemplo, si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez. Determina si el objeto de empate será incluido en la escala de eje Y La barra inicial (coordenada de eje X) donde el objeto de empate será dibujado. Por ejemplo, un valor de 10 pintaría el empate objetan 10 barras atrás. El comienzo y valora la coordenada donde el objeto de empate será dibujado

p408

www.ninjatrader.com

© NinjaTrader, LLC, 2005

endBarsAgo La barra de final (coordenada de eje X) donde el objeto de empate terminará endY El final y valora la coordenada donde el objeto de empate se terminará

Ejemplos

```
//Dibuja un círculo fibonnaci DrawFibonacciCircle ("tag1", verdadero, 10, Bajo [10], 0, Alto [0]);
```

-o-

DrawFibonacciExtensions ()

La definición Dibuja extensiones fibonacci.

DrawFibonacciExtensions de sintaxis (etiqueta de cuerda, bool autoescala, intervalo anchor1BarsAgo, dobles anchor1Value, intervalo anchor2BarsAgo, doblan anchor2Value, intervalo anchor3BarsAgo, doblan anchor3Value,)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta Un usuario definió id único usado para referirse al objeto de empate. Por

ejemplo,

si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez.

autoescala Determina si el objeto de empate será incluido en anchor1BarsAgo de escala
de eje Y El número de hace unas barras (x valor) del 1er punto de ancla anchor1Y
El valor y del 1er punto de ancla anchor2BarsAgo El número de hace unas barras
(x valor) del 2do punto de ancla anchor2Y El valor y del 2do punto de ancla
anchor3BarsAgo El número de hace unas barras (x valor) del 3er punto de ancla
anchor3Y El valor y del 3er punto de ancla

Ejemplos

```
DrawFibonacciExtensions ("tag1", verdadero, 4, Bajo [4], 3, Alto [3], 1, Bajo [1]);
```

-o-

DrawFibonacciRetracements ()

La definición Dibuja fibonacci retracements.

DrawFibonacciRetracements de sintaxis (etiqueta de cuerda, bool autoescala, intervalo startBarsAgo, dobles startY, intervalo endBarsAgo, doblan endY)

Método De Indicador de Encargo, Estrategia de Encargo

p409

www.ninjatrader.com
© NinjaTrader, LLC, 2005

etiqueta

autoescala startBarsAgo

startY endBarsAgo endY

Ejemplos

Los parámetros Un usuario definió id único solían referirse al objeto de

empate. Por ejemplo, si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez. Determina si el objeto de empate será incluido en la escala de eje Y La barra inicial (coordenada de eje X) donde el objeto de empate será dibujado. Por ejemplo, un valor de 10 pintaría el empate objetan 10 barras atrás. El comienzo y valora la coordenada donde el objeto de empate será dibujado La barra de final (coordenada de eje X) donde el objeto de empate terminará El final y coordenada de valor donde el objeto de empate se terminará

```
//Dibuja un fibonacci retracement objetan DrawFibonacciRetracements ("tag1", verdadero, 10, Bajo [10], 0, Alto [0]);
```

-0-

DrawFibonacciTimeExtensions ()

La definición Dibuja extensiones de tiempo fibonacci.

DrawFibonacciTimeExtensions de sintaxis (*etiqueta de cuerda*, *intervalo* *startBarsAgo*, *dobles* *startY*, *intervalo* *endBarsAgo*, *doblan* *endY*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta

autoescala startBarsAgo

startY endBarsAgo endY

Ejemplos

Un usuario definió id único usado para referirse al objeto de empate. Por ejemplo, si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez. Determina si el objeto de empate será incluido en la escala de eje Y La barra inicial (coordenada de eje X) donde el objeto de empate será dibujado. Por ejemplo, un valor de 10 pintaría el empate objetan 10 barras atrás. El comienzo y valora la coordenada donde el objeto de empate será dibujado La barra de final (coordenada de eje X) donde el objeto de empate terminará El final y coordenada de valor donde el objeto de empate se terminará

```
//Dibuja un objeto de extensión de tiempo fibonacci DrawFibonacciTimeExtensions ("tag1", verdadero, 10, Bajo [10], 0, Alto [0]);
```

-o-

DrawGannFan ()

La definición Atrae a un Admirador Gann.

Sintaxis

DrawGannFan (*etiqueta de cuerda*, *bool autoescala*, *intervalo barsAgo*, *doble y*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta	Un usuario definió id único usado para referirse al objeto de empate. Por ejemplo,
	si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez.
autoescala	Determina si el objeto de empate será incluido en barsAgo de escala de eje Y. La barra el objeto será dibujada en. Un valor de 10 sería hace 10 barras. y El valor de y

Ejemplos

```
//Atrae a un Admirador Gann en la barra corriente bajo. DrawGannFan ("tag1", verdadero, 0, Bajo [0]);
```

-o-

DrawHorizontalLine ()

La definición Dibuja una línea horizontal.

Sintaxis **DrawHorizontalLine** (*etiqueta de cuerda*, *doble y*, *color en Color*) **DrawHorizontalLine** (*etiqueta de cuerda*, *bool autoescala*, *doble y*, *color en Color*, *DashStyle dashStyle*, *anchura internacional*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta

autoescala

ycolor

Un usuario definió id único usado para referirse al objeto de empate. Por ejemplo, si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez. Determina si el objeto de empate será incluido en la escala de eje Y. El valor por defecto es [falso](#). Los y valoran El color de objeto de empate ([referencia](#))

p411
<<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frfsysdrawingcolomemberstopic.asp>>
www.ninjatrader.com

dashStyle

DashStyle. Carrera
© NinjaTrader, LLC, 2005

DashStyle. DashDot
DashStyle. DashDotDot
DashStyle. Dot
DashStyle. Sólido
([referencia](#))

anchura La anchura del objeto de empate

Ejemplos

```
//Dibuja una línea horizontal DrawHorizontalLine ("tag1", 1000, Color. Negro);
```

-0-

DrawLine ()

La definición Dibuja una línea entre dos puntos.

DrawLine de sintaxis (*etiqueta de cuerda*, *intervalo startBarsAgo*, *doble startY*, *intervalo endBarsAgo*, *doble endY*, *color en Color*) **DrawLine** (*etiqueta de cuerda*, *bool autoescala*, *intervalo startBarsAgo*, *dobles startY*, *intervalo endBarsAgo*, *doblan endY*, *color en Color*, *DashStyle dashStyle*, *anchura internacional*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta ejemplo,	Un usuario definió id único usado para referirse al objeto de empate. Por si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez.
autoescala	Determina si el objeto de empate será incluido en la escala de eje Y. Falta el valor es falso . startBarsAgo La barra inicial (coordenada de eje X) donde el objeto de empate será dibujado. Por ejemplo, un valor de 10 pintaría el empate objetan 10 barras atrás.
startY	El comienzo y valora la coordenada donde el objeto de empate será dibujado
endBarsAgo	La barra de final (coordenada de eje X) donde el objeto de empate terminará
endY	El final y valora la coordenada donde el objeto de empate terminará el color El color de objeto de empate (referencia)
dashStyle	DashStyle. Carrera DashStyle. DashDot DashStyle. DashDotDot DashStyle. Dot DashStyle. Sólido (referencia)
anchura	La anchura del objeto de empate

p412

www.ninjatrader.co

m

© NinjaTrader, LLC,

2005

Ejemplos

```
//Dibuja una línea verde lima punteada de 10 barras atrás a la barra corriente//con una anchura de 2 pixeles DrawLine ("tag1", falso, 10, 1000, 0, 1001, Color.LimeGreen, DashStyle. Punto, 2);
```

-o-

DrawRay ()

La definición Dibuja una línea que tiene un punto final infinito en una dirección.

DrawRay de sintaxis (*etiqueta de cuerda, intervalo startBarsAgo, doble startY, intervalo endBarsAgo, doble endY, color en Color*) DrawRay (*etiqueta de cuerda, bool autoescala, intervalo startBarsAgo, dobles startY, intervalo endBarsAgo, doblan endY, color en Color, DashStyle dashStyle, anchura internacional*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta	Un usuario definió id único usado para referirse al objeto de empate. Por
----------	---------------------------------------------------------------------------

ejemplo,

si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez.

autoescala Determina si el objeto de empate será incluido en la escala de eje Y. Falta el valor es **falso**.

startBarsAgo La barra inicial (coordenada de eje X) donde el objeto de empate será dibujado. Por ejemplo, un valor de 10 pintaría el empate objetan 10 barras atrás.

startY El comienzo y valora la coordenada donde el objeto de empate será dibujado

endBarsAgo La barra de final (coordenada de eje X) donde el objeto de empate terminará

endY El final y valora la coordenada **donde**

<<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfssystemdrawingcolormemberstopic.asp>> el objeto de empate terminará el color

El color de objeto de empate ([referencia](#))

dashStyle DashStyle. Carrera

DashStyle. DashDot DashStyle. DashDotDot DashStyle. Dot DashStyle. Sólido ([referencia](#))

anchura La anchura del objeto de empate

Ejemplos

```
//Dibuja una línea verde lima de 10 barras atrás por la barra corriente
DrawRay ("tag1", 10, 1000, 0, 1001, Color. LimeGreen);
```

-O-

DrawRectangle ()

La definición Dibuja un rectángulo.

p413

www.ninjatrader.com

© NinjaTrader, LLC,

2005

Sintaxis

DrawRectangle (*etiqueta de cuerda*, *intervalo startBarsAgo*, *doble startY*, *intervalo endBarsAgo*, *doble endY*, *color en Color*) **DrawRectangle** (*etiqueta de cuerda*, *bool autoescala*, *intervalo startBarsAgo*, *dobles startY*, *intervalo endBarsAgo*, *doblan endY*, *color en Color*, *Color backColor*, *opacidad internacional*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta Un usuario definió id único usado para referirse al objeto de empate. Por ejemplo,

si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas

son usadas cada vez, un nuevo objeto de empate será creado cada vez.

autoescala Determina si el objeto de empate será incluido en la escala de eje Y. Falta el valor es **falso**.

startBarsAgo La barra inicial (coordenada de eje X) donde el objeto de empate será dibujado. Por ejemplo, un valor de 10 pintaría el empate objetan 10 barras atrás.

startY El comienzo y valora la coordenada donde el objeto de empate será dibujado

endBarsAgo La barra de final (coordenada de eje X) donde el objeto de empate terminará

endY El final y valora la coordenada donde el objeto de empate terminará el color El color de objeto de empate ([referencia](#)) **backColor** El color llenar para el objeto ([referencia](#))

opacidad Pone el nivel de transparencia para el color llenar. Valores válidos entre 1 - 10.

Ejemplos

```
//Dibuja un rectángulo azul de las 10 barras bajas atrás al nivel más alto de 5
barras atrás DrawRectangle ("tag1", 10, Bajo [10] - TickSize, 5, Alto [5] +
TickSize, Color. Azul);
```

```
//Dibuja un rectángulo azul de las 10 barras bajas atrás al nivel más alto de 5
barras atrás con//un llenar en color o verde pálido con un nivel de transparencia
de 2 DrawRectangle ("tag1", falso, 10, Bajo [10] - TickSize, 5, Alto [5] + TickSize,
Color. PaleGreen, Color. PaleGreen, 2);
```

-o-

DrawRegion ()

La definición Dibuja una región en una carta.

DrawRegion de sintaxis (*etiqueta de cuerda, intervalo startBarsAgo, el intervalo endBarsAgo, serie de IDataSeries, dobla y, Color outlineColor, Color backColor, opacidad internacional*) **DrawRegion** (*etiqueta de cuerda, intervalo startBarsAgo, intervalo endBarsAgo, IDataSeries series1, IDataSeries series2, Color outlineColor, Color backColor, opacidad internacional*)

Método De Indicador de Encargo, Estrategia de Encargo

p414

www.ninjatrader.com
© NinjaTrader, LLC,
2005

etiqueta

startBarsAgo

endBarsAgo

serie, series1, series2 youtlineColor backColor opacidad

Ejemplos

Los parámetros Un usuario definió id único solían referirse al objeto de empate. Por ejemplo, si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez. La barra (coordenada de eje X) donde el punto de partida de la región será dibujado de. Por ejemplo, un valor de 10 pintaría el principio de la región 10 barras atrás. La barra (coordenada de eje X) donde el punto que se termina de la región será dibujado a. Por ejemplo, un valor de 0 pintaría el final en la barra corriente (última barra) de la carta. Cualesquiera DataSeries escriben a máquina el objeto, como un indicador, Cerca, Alto, Bajo etc. El valor del objeto representará un valor de y. Cualquier doble color de contorno de región de valor [\(referencia\)](#) región atrás colorea [\(referencia\)](#) Juegos el nivel de transparencia para el backColor. Valores válidos entre 1 - 10.

```
//Rellena la región entre los grupos bollinger superiores e inferiores DrawRegion  
("tag1", CurrentBar, 0, Bollinger (2, 14).Upper, Bollinger (2, 14).Lower, Color.  
Vacío, Color. Cal, 2);
```

Puntas 1. Pase en color. Vacío al parámetro "outlineColor" si usted no quiere tener un color de contorno. 2. Si usted quisiera llenar una región entre un valor (20 período valor medio móvil simple) y el borde superior de la carta, pasar en un valor extremo al parámetro "y" tal como 1000000.

-o-

DrawRegressionChannel ()

La definición Dibuja un canal de regresión.

DrawRegressionChannel de sintaxis (ensartan la etiqueta, intervalo startBarsAgo, intervalo endBarsAgo, color en Color) DrawRegressionChannel (etiqueta de cuerda, bool autoescala, intervalo startBarsAgo, intervalo endBarsAgo, Color upperColor, DashStyle upperDashStyle, intervalo upperWidth, Color middleColor, DashStyle middleDashStyle, intervalo middleWidth, Color lowerColor, DashStyle lowerDashStyle, intervalo lowerWidth)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta

autoescala

startBarsAgo

endBarsAgo

coloree upperDashStyle, middleDashStyle, lowerDashStyle

upperColor, middleColor, lowerColor upperWidth, middleWidth, lowerWidth

Ejemplos

© NinjaTrader, LLC, 2005

Un usuario definió id único usado para referirse al objeto de empate. Por ejemplo, si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez. Determina si el objeto de empate será incluido en la escala de eje Y. El valor por defecto [es verdad](#). La barra inicial (coordenada de eje X) donde el canal de regresión comenzará La barra de final (coordenada de eje X) usado como el 2do punto de ancla de la regresión canaliza El color de objeto de empate [\(referencia\)](#)

DashStyle. Carrera

DashStyle. DashDot DashStyle. DashDotDot DashStyle. Dot DashStyle. Sólido [\(referencia\)](#)

La línea colorea [\(referencia\)](#)

La anchura de línea

```
//Dibuja un canal de regresión verde que comienza 10 barras atrás  
DrawRegressionChannel ("myTag" + CurrentBar, 10, 0, Color. Verde);
```

DrawSquare ()

La definición Dibuja un cuadrado.

DrawSquare de sintaxis (*etiqueta de cuerda*, *bool autoescala*, *intervalo barsAgo*, *doble y*, *color en Color*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta

autoescale barsAgo ycolor

Un usuario definió id único usado para referirse al objeto de empate. Por ejemplo, si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez. Determina si el objeto de empate será incluido en la escala de eje Y La barra el objeto será dibujado en. Un valor de 10 sería hace 10 barras. Los y valoran El color de objeto de empate ([referencia](#))

p416

www.ninjatrader.com

© NinjaTrader, LLC,

2005

Ejemplos

```
//Pinta una Plaza Roja en la barra corriente 1 marca debajo de DrawSquare bajo  
("tag1", verdadero, 0, Bajo [0] - TickSize, Color. Rojo);
```

-0-

DrawText ()

La definición Dibuja el texto.

DrawText de sintaxis (*etiqueta de cuerda*, *texto de cuerda*, *intervalo barsAgo*, *doble y*, *color en Color*) **DrawText** (*etiqueta de cuerda*, *bool autoescala*, *el texto de cuerda*, *intervalo barsAgo*, *doble y*, *intervalo yPixelOffset*, *Color textColor*, *fuentes de Fuente*, *alineación de StringAlignment*, *Color outlineColor*, *Color backColor*, *opacidad internacional*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta
ejemplo,

Un usuario definió id único usado para referirse al objeto de empate. Por

si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez.

autoescala Determina si el objeto de empate será incluido en la escala de eje Y. Falta el valor es **falso**. **texto** El texto usted desea dibujar **barsAgo** La barra (coordenada de eje X) donde el objeto de empate será dibujado. Para ejemplo, un valor de 10 pintaría el empate objetan 10 barras atrás. y La posición de coordenada de y el objeto será dibujada **yPixelOffset** El valor de compensación en píxeles del **especificado**

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfssystemdrawingcolormemberstopic.asp> y valor de precios **textColor** El color de texto de objeto de empate [\(referencia\)](#)

fuentes El tipo de fuente usó [\(referencia\)](#)

alineación StringAlignment. Centro
StringAlignment. StringAlignment lejano. Cerca de [\(referencia\)](#)

outlineColor Color de contorno de cuadro de texto [\(referencia\)](#)

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfssystemdrawingcolormemberstopic.asp>

backColor El cuadro de texto atrás colorea [\(referencia\)](#) la opacidad **opacity** Pone el nivel de transparencia para el color llenar. Valores válidos entre 1 - 10.

Ejemplos

```
//Dibuja texto
DrawText ("tag1", "Texto para dibujar", 10, 1000, Color.Negro);
```

-O-

DrawTextFixed ()

p417

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Definición

Dibuja el texto en una de 5 posiciones fijas predefinidas disponibles en el panel 1 (panel de precios) de una carta.

DrawTextFixed de sintaxis (*ensartan la etiqueta, el texto de cuerda, TextPosition textPosition*) **DrawTextFixed** (*etiqueta de cuerda, texto de cuerda, TextPosition textPosition, Color textColor, fuente de Fuente, Color outlineColor, Color backColor, opacidad internacional*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta ejemplo, Un usuario definió id único usado para referirse al objeto de empate. Por si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez.

texto El texto usted desea atraer TextPosition TextPosition. BottomLeft TextPosition. BottomRight TextPosition. Centro TextPosition. TopLeft TextPosition. TopRight

textColor El color de texto de objeto de empate [\(referencia\)](#)

fuente El tipo de fuente usó [\(referencia\)](#) outlineColor Color de contorno de cuadro de texto_ [\(referencia\)](#)

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfssystemdrawingcolormemberstopic.asp>

backColor El cuadro de texto atrás colorea [\(referencia\)](#) la opacidad Pon el nivel de transparencia para el color llenar. Valores válidos entre 1 - 10.

Ejemplos

```
//Dibuja el texto en la esquina correcta superior del panel 1
DrawTextFixed ("tag1", "Texto para atraer", TextPosition. TopRight);
```

-o-

DrawTriangleDown ()

La definición Dibuja un punto de triángulo abajo.

DrawTriangleDown de sintaxis (*etiqueta de cuerda, bool autoescala, intervalo barsAgo, doble y, color en Color*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

p418

www.ninjatrader.co

m

etiqueta

autoescale barsAgo ycolor

Ejemplos

© NinjaTrader, LLC, 2005

Un usuario definió id único usado para referirse al objeto de empate. Por ejemplo, si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez. Determina si el objeto de empate será incluido en la escala de eje Y La barra el objeto será dibujado en. Un valor de 10 sería hace 10 barras. Los y valoran El color de objeto de empate ([referencia](#))

```
//Pinta un triángulo rojo que señala abajo en la barra corriente 1 marca debajo de  
DrawTriangleDown bajo ("tag1", verdadero, 0, Bajo [0] - TickSize, Color. Rojo);
```

-0-

DrawTriangleUp ()

La definición Prepara un punto de triángulo.

DrawTriangleUp de sintaxis (*etiqueta de cuerda*, *bool autoescale*, *intervalo barsAgo*, *doble y*, *color en Color*)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta

autoescale barsAgo ycolor

Ejemplos

Un usuario definió id único usado para referirse al objeto de empate. Por ejemplo, si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez. Determina si el objeto de empate será incluido en la escala de eje Y La barra el objeto será dibujado en. Un valor de 10 sería hace 10 barras. Los y valoran El color de objeto de empate ([referencia](#))

```
//Pinta un triángulo rojo que señala en la barra corriente 1 marca debajo de  
DrawTriangleUp bajo ("tag1", verdadero, 0, Bajo [0] - TickSize, Color. Rojo);
```

-0-

DrawVerticalLine ()

La definición Dibuja una línea vertical.

DrawVerticalLine de sintaxis (ensartan la etiqueta, intervalo barsAgo, color en Color)

DrawVerticalLine (etiqueta de cuerda, intervalo barsAgo, color en Color, DashStyle dashStyle, anchura internacional)

p419

www.ninjatrader.com

© NinjaTrader, LLC, 2005

Método De Indicador de Encargo, Estrategia de Encargo

Parámetro

s

etiqueta

los barsAgo colorean dashStyle

anchura

Ejemplos

Un usuario definió id único usado para referirse al objeto de empate. Por ejemplo, si usted pasa en un valor de "myTag", cada vez que esta etiqueta es usada, mismo dibuja el objeto es modificado. Si las etiquetas únicas son usadas cada vez, un nuevo objeto de empate será creado cada vez. La barra el objeto será dibujada en. Un valor de 10 sería hace 10 barras. El color de objeto de empate [\(referencia\)](#)

DashStyle. Carrera

DashStyle. DashDot DashStyle. DashDotDot DashStyle. Dot DashStyle. Sólido [\(referencia\)](#)

La anchura del objeto de empate

```
//Dibuja una línea horizontal DrawVerticalLine  
("tag1", 10, Color. Negro);
```

-o-

RemoveDrawObject ()

La definición Quita un objeto de empate de la carta basada en su valor de etiqueta.

`RemoveDrawObject` **de sintaxis** (ensartan la etiqueta)

Método De Indicador de Encargo, Estrategia de Encargo

Parámetros

etiqueta Un usuario definió el valor de id único que fue pasado en cualquier del
empate métodos. Esto se identificará qué objeto tiene que ser quitado de la
carta.

Ejemplos

```
//Quita un objeto de empate con la etiqueta "tag1"  
RemoveDrawObject ("tag1");
```

-o-

RemoveDrawObjects ()

La definición Quita a todos dibujan el objeto de la carta.

p420

www.ninjatrader.co

m

Sintaxis `RemoveDrawObjects ()`

Método De Indicador de Encargo, Estrategia de Encargo

Ejemplos

```
//Quita a todos dibujan objetos RemoveDrawObjects ();  
© NinjaTrader, LLC, 2005
```

Métodos de Indicador compartidos

Descripción

La referencia de indicador proporciona definiciones, sintaxis, definiciones de parámetro y ejemplos para NinjaTrader métodos de indicador internos.

Datos de Entrada válidos para Métodos de

Indicador

El método de indicador puede aceptar las formas válidas siguientes de datos de entrada.

La entrada de Falta **de Entrada de falta** del indicador de encargo, Mercado la columna de Analizador o la estrategia son usadas como la entrada de falta si la entrada no es especificada.

```
//La imprenta del valor corriente del 10 período SMA de cotizaciones de
cierre//utilización de la falta introducida. doblo el valor = SMA (10) [0]; la Letra
("El valor de SMA corriente es" + valor. ToString ());
```

La Serie de precios Abierta, Alto, Bajo, Cerca y Volumen puede ser todo usada como introducido para un método de indicador.

```
//El paso en una serie de precios de Altos precios e impresión el valor corriente
del//14 período valor medio móvil simple dobla el valor = SMA (Alto, 14) [0]; la Letra
("El valor de SMA corriente es" + valor. ToString ());
```

Los Indicadores **de indicador** pueden ser usados como introducido para indicadores.

```
//La imprenta del valor corriente del 20 período valor medio móvil simple de un 14
período RSI//utilización de una serie de datos de cotizaciones de cierre dobla el valor
= SMA (RSI (Cerca, 14), 20) [0]; la Letra ("El valor de SMA corriente es" + valor.
ToString ());
```

DataSet [DataSet](#) puede ser usado como introducido para indicadores.

```
//Instantiating un nuevo objeto de DataSet y paso de esto en como introducido
para calcular//un valor medio móvil simple DataSet myDataSet = nuevo DataSet
(esto); doblo el valor = SMA (myDataSet, 20) [0];
```

El Objeto de barras Un objeto de Barras (sostiene una serie del objeto de barra que contienen datos OHLC) puede ser usado como introducido para indicadores.

```
//El paso en el segundo objeto de Barras sostenido en un multiinstrumento y estrategia de margen de tiempo//El valor por defecto usado para el cálculo SMA es el precio cercano  
doble valor = SMA (BarsArray [1], 20) [0]; la Letra ("El valor de SMA corriente es"  
+ valor. ToString ());
```

-0-

Aroon

Cortesía de definición de www.stockcharts.com

Sintaxis Aroon (*período internacional*) Aroon (IDataSeries inputData, *período internacional*)

Las vueltas valoran Aroon (*período internacional*).Up [*intervalo barsAgo*] Aroon (IDataSeries inputData, *período internacional*).Up [*intervalo barsAgo*]

Las vueltas abajo valoran Aroon (*período internacional*).Down [*intervalo barsAgo*] Aroon (IDataSeries inputData, *período internacional*).Down [*intervalo barsAgo*]

Devuelva el Tipo *doble*; Teniendo acceso a este método vía un valor de índice [*el intervalo barsAgo*] devuelve el valor de indicador de la barra referida.

Parámetros

período El número de barras para incluir en el cálculo inputData

Fuente de datos(?) de indicador

Ejemplos

```
//Imprime la corriente/abajo valores de un 20 período Aroon usando el tipo de precios de falta  
doblan upValue = Aroon (20).Up [0]; doble downValue = Aroon (20).Down [0];  
la Letra ("Aroon corriente valoran es" + upValue. ToString ()); la Letra ("Aroon corriente abajo valoran es" + downValue. ToString ());
```

p422

www.ninjatrader.com
© NinjaTrader, LLC,

2005

Código fuente

Usted puede abrir el código fuente de indicador vía el Redactor de NinjaScript.

-O-

Línea de acumulación/Distribución (ADL)

[Cortesía de definición de www.stockcharts.com](http://www.stockcharts.com)

Sintaxis ADL () ADL (IDataSeries inputData)

ADL () [intervalo barsAgo] ADL (IDataSeries inputData) [intervalo barsAgo]

Devuelva el Tipo doble; Teniendo acceso a este método vía un valor de índice [el intervalo barsAgo] devuelve el valor de indicador de la barra referida.

Parámetros inputData Fuente de datos(?) de
indicador

Ejemplos

```
//Los controles si ADL se eleva bool isRising = Elevándose (ADL); la Letra ("¿Es la  
rebelión de ADL?" + isRising);
```

Código fuente Usted puede abrir el código fuente de indicador vía el Redactor de NinjaScript.

-O-

Zona de Precios adaptable (APZ)

Definición El indicador Adaptive Price Zone del S&C, el artículo "Trading With An Adaptive Price Zone" de septiembre de 2006 por Lee Leibfarth es un juego de grupos basados en un doble valor medio móvil exponencial liso a corto plazo. Los grupos forman un canal que rodea el precio medio y rastrea fluctuaciones de precios rápidamente, sobre todo en mercados volátiles. Como el precio se cruza encima de la zona que puede señalar una oportunidad de vender en previsión de una inversión. Como el precio se cruza debajo de la zona que puede señalar una oportunidad de comprar en previsión de una inversión.

APZ de sintaxis (doblan barPct, período internacional) APZ (IDataSeries inputData, doble barPct, período internacional)

Vuelve valor de grupo superior APZ (doble barPct, período internacional).Upper [intervalo barsAgo] APZ (IDataSeries inputData, doble barPct, período internacional).Upper [intervalo barsAgo]

p423

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Las vueltas bajan el valor de grupo

APZ (doblan *barPct*, período internacional).Lower [intervalo *barsAgo*] APZ (IDataSeries *inputData*, doble *barPct*, período internacional).Lower [intervalo *barsAgo*]

Devuelva el Tipo *doble*; Teniendo acceso a este método vía un valor de índice [el intervalo *barsAgo*] devuelve el valor de indicador de la barra referida.

Parámetros

barPct El número de período de desviaciones estándar El número de barras para incluir en el cálculo *inputData* Fuente de datos(?) de indicador

Ejemplos

```
//Imprime el valor de grupo superior corriente de un 20 período APZ
doblan upperValue = APZ (2, 20).Upper [0]; la Letra ("El valor superior
APZ corriente es" + upperValue. ToString ());
```

Código fuente Usted puede abrir el código fuente de indicador vía el Redactor de NinjaScript.

-o-

Oscilador de Aroon

Cortesía **de definición** de www.stockcharts.com

Sintaxis AroonOscillator (*período* internacional) AroonOscillator (IDataSeries *inputData*, período internacional)

Valor por defecto de vueltas AroonOscillator (*período* internacional) [intervalo *barsAgo*] AroonOscillator (IDataSeries *inputData*, período internacional) [intervalo *barsAgo*]

Devuelva el Tipo *doble*; Teniendo acceso a este método vía un valor de índice [el intervalo *barsAgo*] devuelve el valor de indicador de la barra referida.

Parámetros

período El número de barras para incluir en el cálculo *inputData* Fuente de datos(?) de indicador

Ejemplos

```
//Imprime los valores corrientes de un 20 período AroonOscillator usando el tipo de precios de falta doblan upValue = AroonOscillator (20) [0]; la Letra ("El valor de AroonOscillator corriente es" + upValue. ToString ());  
  
//Imprime los valores corrientes de un 20 período AroonOscillator usando el tipo de alto precio doblan upValue = AroonOscillator (Alto, 20) [0]; la Letra ("El valor de AroonOscillator corriente es" + upValue. ToString ());
```

Código fuente Usted puede abrir el código fuente de indicador vía el Redactor de NinjaScript.

-0-

Índice Direccional medio (ADX)

[Cortesía de definición de www.stockcharts.com](http://www.stockcharts.com)

Sintaxis ADX (*período internacional*) ADX (IDataSeries *inputData*, *período internacional*)

Valor por defecto de vueltas ADX (*período internacional*) [*intervalo barsAgo*] ADX (IDataSeries *inputData*, *período internacional*) [*intervalo barsAgo*]

Devuelva el Tipo **doble**; Teniendo acceso a este método vía un valor de índice [**el intervalo barsAgo**] devuelve el valor de indicador de la barra referida.

Parámetros

período El número de barras para incluir en el cálculo **inputData**
Fuente de datos(?) de indicador

Ejemplos

```
//Imprime el valor corriente de un 20 período ADX utilización del tipo de precios de falta doble valor = ADX (20) [0]; la Letra ("El valor de ADX corriente es" + valor. ToString ());  
  
//Imprime el valor corriente de un 20 período ADX que usa el tipo de alto precio doble valor = ADX (Alto, 20) [0]; la Letra ("El valor de ADX corriente es" + valor. ToString ());
```

Código fuente Usted puede abrir el código fuente de indicador vía el Redactor de NinjaScript.

-0-

Movimiento Direccional medio que Tasa (ADXR)

La definición El ADXR es igual a ADX corriente más el ADX de hace barras unas n dividido en 2.

Sintaxis `ADXR (intervalo internacional, período internacional) ADXR (IDataSeries inputData, intervalo internacional, período internacional)`

Valor por defecto de vueltas `ADXR (intervalo internacional, período internacional) [intervalo barsAgo] ADXR (IDataSeries inputData, intervalo internacional, período internacional) [intervalo barsAgo]`

Devuelva el Tipo `doble`; Teniendo acceso a este método vía un valor de índice `[el intervalo barsAgo]` devuelve el valor de indicador de la barra referida.

Parámetros

intervalo	El intervalo entre el primer valor de ADX y ADX corriente valora el período
El número de barras para incluir en el cálculo	<code>inputData</code> Fuente de datos(?)
de indicador	

Ejemplos

```
//Imprime el valor corriente de un 20 período ADXR utilización del tipo de
precios de falta doble valor = ADXR (10, 20) [0]; la Letra ("El valor de ADXR
corriente es" + valor.ToString ());
```

Código fuente Usted puede abrir el código fuente de indicador vía el Redactor de NinjaScript.

-o-

Variedad Verdadera media (ATR)

Cortesía **de definición** de www.stockcharts.com

<<http://www.marketscreen.com/help/atoz/default.asp?hideHF=&Num=28>> Courtesy de www.marketscreen.com

Sintaxis `ATR (período internacional) ATR (IDataSeries inputData, período internacional)`

Valor por defecto de vueltas `ATR (período internacional) [intervalo barsAgo] ATR (IDataSeries inputData, período internacional) [intervalo barsAgo]`

Devuelva el Tipo *doble*; Teniendo acceso a este método vía un valor de índice [*el intervalo barsAgo*] devuelve el valor de indicador de la barra referida.

p426

www.ninjatrader.com
© NinjaTrader, LLC, 2005

Período **de parámetros**

El número de barras para incluir en el cálculo inputData

Fuente de datos(?) de indicador

Ejemplos

```
//Imprime el valor corriente de un 20 período ATR utilización del tipo de  
precios de falta doble valor = ATR (20) [0]; la Letra ("El valor de ATR  
corriente es" + valor. ToString ());
```

Código fuente Usted puede abrir el código fuente de indicador vía el Redactor de NinjaScript.

-o-

Equilibrio de fuerzas (BOP)

La definición El equilibrio de fuerzas (BOP) indicador mide la fuerza de los toros contra osos tasando la capacidad de cada uno de empujar el precio a un nivel extremo.

BOP de sintaxis (*intervalo* liso) BOP (IDataSeries *inputData*, *intervalo* liso)

BOP de valor por defecto de vueltas (*intervalo* liso) [*intervalo barsAgo*] BOP (IDataSeries *inputData*, *intervalo* liso) [*intervalo barsAgo*]

Devuelva el Tipo *doble*; Teniendo acceso a este método vía un valor de índice [*el intervalo barsAgo*] devuelve el valor de indicador de la barra referida.

Parámetros

liso El período de allanamiento
inputData Fuente de datos(?) de
indicador

Ejemplos

```
//Imprime el valor corriente del BOP usando el tipo de precios de falta y 3  
allanamiento de período doble valor = BOP (3) [0]; la Letra ("El valor de BOP  
corriente es" + valor. ToString ());
```

Código fuente Usted puede abrir el código fuente de indicador vía
el Redactor de NinjaScript.

-O-

p427

www.ninjatrader.com