

```
input marketOpen = 930;
input marketClose = 1600;
input intraDaySpan = {Default "SameDay" , "OverNight"};
input alertOnBreak = yes;
input alertOnPullBack = yes;
input numberOfDays = 0;
input numberOfYears = 0;
input showFibLabels = yes;
input plotFib1 = yes;
input fibCoefficient1 = -1.618;
input plotFib2 = yes;
input fibCoefficient2 = -1.236;
input plotFib3 = yes;
input fibCoefficient3 = -1.0;
input plotFib4 = yes;
input fibCoefficient4 = -0.5;
input plotFib5 = yes;
input fibCoefficient5 = 0.5;
input plotFib6 = yes;
input fibCoefficient6 = 0.618;
input plotFib7 = yes;
input fibCoefficient7 = 1.5;
input plotFib8 = yes;
input fibCoefficient8 = 2.0;
input plotFib9 = yes;
input fibCoefficient9 = 2.236;
input plotFib10 = yes;
input fibCoefficient10 = 2.618;
input plotFib11 = yes;
input fibCoefficient11 = -1.5;
input plotFib12 = yes;
input fibCoefficient12 = -1.382;
input plotFib13 = yes;
input fibCoefficient13 = 0.382;
input plotFib14 = yes;
input fibCoefficient14 = 1.618;
```

```
def okToPlot = GetLastDay() - numberOfDays <= GetDay() and GetLastYear() -
numberOfYears <= GetYear() ;
```

```
def regularSessionHours = RegularTradingStart(GetYYYYMMDD()) <= GetTime();
def extendedSessionHours = RegularTradingStart(GetYYYYMMDD()) >= GetTime();
```

```
def extendedSessionStart = regularSessionHours[1] and extendedSessionHours;
def regularSessionStart = extendedSessionHours[1] and regularSessionHours;
```

```
def extendedSessionHigh = CompoundValue(1, if extendedSessionStart then high else if
extendedSessionHours then Max(high, extendedSessionHigh[1]) else extendedSessionHigh[1],
0);
```

```
def extendedSessionLow = CompoundValue(1, if extendedSessionStart then low else if
extendedSessionHours then Min(low, extendedSessionLow[1]) else extendedSessionLow[1], 0);
```

```
def regularSessionHigh = CompoundValue(1, if regularSessionStart then high else if
regularSessionHours then Max(high, regularSessionHigh[1]) else regularSessionHigh[1], 0);
```

```
def regularSessionLow = CompoundValue(1, if regularSessionStart then low else if
regularSessionHours then Min(low, regularSessionLow[1]) else regularSessionLow[1], 0);
```

```
#----- Plot the overnight high/low during regular market hours
plot overnightHigh = if okToPlot and regularSessionHours then extendedSessionHigh else
Double.NaN;
overnightHigh.SetDefaultColor(Color.DARK_GREEN);
overnightHigh.SetLineWeight(3);
plot overnightLow = if okToPlot and regularSessionHours then extendedSessionLow else
Double.NaN;
overnightLow.SetDefaultColor(Color.DARK_RED);
overnightLow.SetLineWeight(3);
```

```
def breakOfHigh = if regularSessionStart then low < extendedSessionHigh and high >
extendedSessionHigh else extendedSessionHigh[1] > regularSessionHigh[1] and
regularSessionHigh > extendedSessionHigh;
def breakOfLow = if regularSessionStart then high > extendedSessionLow and low <
extendedSessionLow else extendedSessionLow[1] < regularSessionLow[1] and
regularSessionLow < extendedSessionLow;
```

```
#----- Throw some alerts in here
Alert(alertOnBreak and breakOfHigh, "Break Above Overnight High", Alert.BAR,
Sound.RING);
Alert(alertOnBreak and breakOfLow, "Break Below Overnight Low", Alert.BAR,
Sound.RING);
```

```
def pullbackToHigh = low[1] > extendedSessionHigh[1] and low < extendedSessionHigh;
def pullbackToLow = high[1] < extendedSessionLow[1] and high > extendedSessionLow;
```

```
#----- Throw some alerts in here
Alert(alertOnPullBack and pullbackToHigh, "Pullback To Overnight High", Alert.BAR,
Sound.RING);
Alert(alertOnPullBack and pullbackToLow, "Pullback To Overnight Low", Alert.BAR,
Sound.RING);
```

```

#----- Plot some arrows here
plot breakOfHighArrow = if okToPlot and alertOnBreak and breakOfHigh then low else
Double.NaN;
breakOfHighArrow.SetDefaultColor(Color.CYAN);
breakOfHighArrow.SetPaintingStrategy(PaintingStrategy.ARROW_UP);
breakOfHighArrow.HideBubble();
breakOfHighArrow.HideTitle();
plot breakOfLowArrow = if okToPlot and alertOnBreak and breakOfLow then high else
Double.NaN;
breakOfLowArrow.SetDefaultColor(Color.MAGENTA);
breakOfLowArrow.SetPaintingStrategy(PaintingStrategy.ARROW_DOWN);
breakOfLowArrow.HideBubble();
breakOfLowArrow.HideTitle();

plot pullbackToHighArrow = if okToPlot and alertOnPullBack and pullbackToHigh then high
else Double.NaN;
pullbackToHighArrow.SetDefaultColor(Color.MAGENTA);
pullbackToHighArrow.SetPaintingStrategy(PaintingStrategy.ARROW_DOWN);
plot pullbackToLowArrow = if okToPlot and alertOnPullBack and pullbackToLow then low else
Double.NaN;
pullbackToLowArrow.SetDefaultColor(Color.CYAN);
pullbackToLowArrow.SetPaintingStrategy(PaintingStrategy.ARROW_UP);

#----- Draw some fib retracements and extensions
def overnightRange = overnightHigh - overnightLow;
plot fibLevel1;
fibLevel1.SetDefaultColor(Color.WHITE);
fibLevel1.SetPaintingStrategy(PaintingStrategy.DASHES);
fibLevel1.HideBubble();
fibLevel1.HideTitle();
plot fibLevel2;
fibLevel2.SetDefaultColor(Color.WHITE);
fibLevel2.SetPaintingStrategy(PaintingStrategy.DASHES);
fibLevel2.HideBubble();
fibLevel2.HideTitle();
plot fibLevel3;
fibLevel3.SetDefaultColor(Color.WHITE);
fibLevel3.SetPaintingStrategy(PaintingStrategy.DASHES);
fibLevel3.HideBubble();
fibLevel3.HideTitle();
plot fibLevel4;
fibLevel4.SetDefaultColor(Color.WHITE);
fibLevel4.SetPaintingStrategy(PaintingStrategy.DASHES);
fibLevel4.HideBubble();
fibLevel4.HideTitle();
plot fibLevel5;

```

```
fibLevel5.SetDefaultColor(Color.WHITE);
fibLevel5.SetPaintingStrategy(PaintingStrategy.DASHES);
fibLevel5.HideBubble();
fibLevel5.HideTitle();
plot fibLevel6;
fibLevel6.SetDefaultColor(Color.WHITE);
fibLevel6.SetPaintingStrategy(PaintingStrategy.DASHES);
fibLevel6.HideBubble();
fibLevel6.HideTitle();
plot fibLevel7;
fibLevel7.SetDefaultColor(Color.WHITE);
fibLevel7.SetPaintingStrategy(PaintingStrategy.DASHES);
fibLevel7.HideBubble();
fibLevel7.HideTitle();
plot fibLevel8;
fibLevel8.SetDefaultColor(Color.WHITE);
fibLevel8.SetPaintingStrategy(PaintingStrategy.DASHES);
fibLevel8.HideBubble();
fibLevel8.HideTitle();
plot fibLevel9;
fibLevel9.SetDefaultColor(Color.WHITE);
fibLevel9.SetPaintingStrategy(PaintingStrategy.DASHES);
fibLevel9.HideBubble();
fibLevel9.HideTitle();
plot fibLevel10;
fibLevel10.SetDefaultColor(Color.WHITE);
fibLevel10.SetPaintingStrategy(PaintingStrategy.DASHES);
fibLevel10.HideBubble();
fibLevel10.HideTitle();
plot fibLevel11;
fibLevel11.SetDefaultColor(Color.WHITE);
fibLevel11.SetPaintingStrategy(PaintingStrategy.DASHES);
fibLevel11.HideBubble();
fibLevel11.HideTitle();
plot fibLevel12;
fibLevel12.SetDefaultColor(Color.WHITE);
fibLevel12.SetPaintingStrategy(PaintingStrategy.DASHES);
fibLevel12.HideBubble();
fibLevel12.HideTitle();
plot fibLevel13;
fibLevel13.SetDefaultColor(Color.WHITE);
fibLevel13.SetPaintingStrategy(PaintingStrategy.DASHES);
fibLevel13.HideBubble();
fibLevel13.HideTitle();
plot fibLevel14;
fibLevel14.SetDefaultColor(Color.WHITE);
```

```
fibLevel14.SetPaintingStrategy(PaintingStrategy.DASHES);
fibLevel14.HideBubble();
fibLevel14.HideTitle();
```

```
fibLevel1 = if plotFib1 and okToPlot and regularSessionHours then overnightHigh -
(overnightRange * fibCoefficient1) else Double.NaN ;
fibLevel2 = if plotFib2 and okToPlot and regularSessionHours then overnightHigh -
(overnightRange * fibCoefficient2) else Double.NaN ;
fibLevel3 = if plotFib3 and okToPlot and regularSessionHours then overnightHigh -
(overnightRange * fibCoefficient3) else Double.NaN ;
fibLevel4 = if plotFib4 and okToPlot and regularSessionHours then overnightHigh -
(overnightRange * fibCoefficient4) else Double.NaN ;
fibLevel5 = if plotFib5 and okToPlot and regularSessionHours then overnightHigh -
(overnightRange * fibCoefficient5) else Double.NaN ;
fibLevel6 = if plotFib6 and okToPlot and regularSessionHours then overnightHigh -
(overnightRange * fibCoefficient6) else Double.NaN ;
fibLevel7 = if plotFib7 and okToPlot and regularSessionHours then overnightHigh -
(overnightRange * fibCoefficient7) else Double.NaN ;
fibLevel8 = if plotFib8 and okToPlot and regularSessionHours then overnightHigh -
(overnightRange * fibCoefficient8) else Double.NaN ;
fibLevel9 = if plotFib9 and okToPlot and regularSessionHours then overnightHigh -
(overnightRange * fibCoefficient9) else Double.NaN ;
fibLevel10 = if plotFib10 and okToPlot and regularSessionHours then overnightHigh -
(overnightRange * fibCoefficient10) else Double.NaN ;
```

```
fibLevel11 = if plotFib1 and okToPlot and regularSessionHours then overnightHigh -
(overnightRange * fibCoefficient1) else Double.NaN ;
fibLevel12 = if plotFib2 and okToPlot and regularSessionHours then overnightHigh -
(overnightRange * fibCoefficient2) else Double.NaN ;
fibLevel13 = if plotFib3 and okToPlot and regularSessionHours then overnightHigh -
(overnightRange * fibCoefficient3) else Double.NaN ;
fibLevel14 = if plotFib4 and okToPlot and regularSessionHours then overnightHigh -
(overnightRange * fibCoefficient4) else Double.NaN ;
```

```
#----- Add chart bubbles to label each fib level
```

```
DefineGlobalColor("Fib Labels", Color.WHITE);
AddChartBubble(showFibLabels and plotFib1 and regularSessionStart, fibLevel1,
fibCoefficient1, GlobalColor("Fib Labels"), no);
AddChartBubble(showFibLabels and plotFib2 and regularSessionStart, fibLevel2,
fibCoefficient2, GlobalColor("Fib Labels"), no);
AddChartBubble(showFibLabels and plotFib3 and regularSessionStart, fibLevel3,
fibCoefficient3, GlobalColor("Fib Labels"), no);
AddChartBubble(showFibLabels and plotFib4 and regularSessionStart, fibLevel4,
fibCoefficient4, GlobalColor("Fib Labels"), no);
AddChartBubble(showFibLabels and plotFib5 and regularSessionStart, fibLevel5,
fibCoefficient5, GlobalColor("Fib Labels"), no);
```

```

AddChartBubble(showFibLabels and plotFib6 and regularSessionStart, fibLevel6,
fibCoefficient6, GlobalColor("Fib Labels"), no);
AddChartBubble(showFibLabels and plotFib7 and regularSessionStart, fibLevel7,
fibCoefficient7, GlobalColor("Fib Labels"), no);
AddChartBubble(showFibLabels and plotFib8 and regularSessionStart, fibLevel8,
fibCoefficient8, GlobalColor("Fib Labels"), no);
AddChartBubble(showFibLabels and plotFib9 and regularSessionStart, fibLevel9,
fibCoefficient9, GlobalColor("Fib Labels"), no);
AddChartBubble(showFibLabels and plotFib10 and regularSessionStart, fibLevel10,
fibCoefficient10, GlobalColor("Fib Labels"), no);

def OpenCounter = SecondsFromTime(marketOpen);
def CloseCounter = SecondsTillTime(marketClose);

def MarketHours = if OpenCounter >= 0 and CloseCounter >= 0 then 1 else 0;

def beforeMidnight = OpenCounter >= 0 and CloseCounter <= 0;
def afterMidnight = OpenCounter <= 0 and CloseCounter >= 0 ;

def Today ;
def hideChartBubbles ;
rec DailyHigh ;
rec DailyLow ;

switch (intraDaySpan) {
case "SameDay":
Today = if GetDay() != GetDay()[1] then 1 else 0;
DailyHigh = if MarketHours then if high > DailyHigh[1] then high else DailyHigh[1] else high;
DailyLow = if Today then low else if MarketHours then if low < DailyLow[1] then low else
DailyLow[1] else low;
hideChartBubbles = MarketHours;
case "OverNight":
Today = 0;
DailyHigh = if beforeMidnight or afterMidnight then if high > DailyHigh[1] then high else
DailyHigh[1] else high;
DailyLow = if beforeMidnight or afterMidnight then if low < DailyLow[1] then low else
DailyLow[1] else low;
hideChartBubbles = beforeMidnight or afterMidnight;
};

plot TodaysHigh = if okToPlot and hideChartBubbles then DailyHigh else Double.NaN;
plot TodaysLow = if okToPlot and hideChartBubbles then DailyLow else Double.NaN;
TodaysHigh.SetDefaultColor(Color.GREEN);
TodaysLow.SetDefaultColor(Color.RED);

Alert(DailyHigh > DailyHigh[1] and hideChartBubbles, "New High", Alert.BAR, Sound.RING);

```

```
Alert(DailyLow < DailyLow[1] and hideChartBubbles, "New Low", Alert.BAR, Sound.RING);
```