



ArraySetAsSeries

The function sets the AS_SERIES flag to a selected [object of a dynamic array](#), and elements will be indexed like in [timeseries](#).

```
bool ArraySetAsSeries(
    const void& array[], // array by reference
    bool flag            // true denotes reverse order of indexing
);
```

Parameters

array[]
[in][out] Numeric array to set.

flag
[in] Array indexing direction.

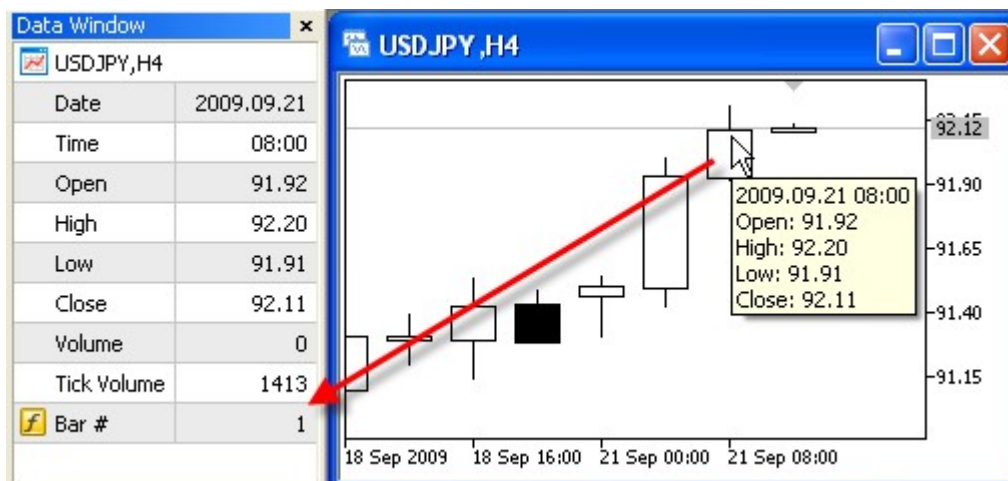
Return Value

The function returns true on success, otherwise - false.

Note

The [AS_SERIES](#) flag can't be set for multi-dimensional arrays or static arrays (arrays, whose size in square brackets is preset already on the compilation stage). Indexing in timeseries differs from a common array in that the elements of timeseries are indexed from the end towards the beginning (from the newest to oldest data).

Example: Indicator that shows bar number



```

#property indicator_chart_window
#property indicator_buffers 1
//--- indicator buffers
double      NumerationBuffer[];
//+-----+
//| Custom indicator initialization function |
//+-----+
int OnInit()
{
//--- indicator buffers mapping
    SetIndexBuffer(0,NumerationBuffer,INDICATOR_DATA);
//--- set buffer style
    SetIndexStyle(0,DRAW_LINE,STYLE_SOLID,1,CLR_NONE);
//--- set indexing for the buffer like in timeseries
    ArraySetAsSeries(NumerationBuffer,true);
//--- set accuracy of showing in DataWindow
    IndicatorSetInteger(INDICATOR_DIGITS,0);
//--- how the name of the indicator array is displayed in DataWindow
    IndicatorShortName("Bar #");
//---
    return(INIT_SUCCEEDED);
}
//+-----+
//| Custom indicator iteration function |
//+-----+
int OnCalculate(const int rates_total,
               const int prev_calculated,
               const datetime &time[],
               const double &open[],
               const double &high[],
               const double &low[],
               const double &close[],
               const long &tick_volume[],
               const long &volume[],
               const int &spread[])
{
//--- we'll store the time of the current zero bar opening
    static datetime currentBarTimeOpen=0;
//--- revert access to array time[] - do it like in timeseries
    ArraySetAsSeries(time,true);
//--- If time of zero bar differs from the stored one
    if(currentBarTimeOpen!=time[0])
    {
        //--- enumerate all bars from the current to the chart depth
        for(int i=rates_total-1;i>=0;i--) NumerationBuffer[i]=i;
        currentBarTimeOpen=time[0];
    }
//--- return value of prev_calculated for next call
    return(rates_total);
}

```

See also

[Access to timeseries](#), [ArrayGetAsSeries](#)