

```
#ifndef _ACSILDEPTHBARS_H_
#define _ACSILDEPTHBARS_H_
```

```
#include "scstructures.h"
```

```
class c_InternalACSILDepthBars;
```

```
/******
```

```
// c_ACSILDepthBars
```

This class provides an interface for accessing historical market depth data bars from a single chart. Use `sc.GetMarketDepthBars` or `sc.GetMarketDepthBarsFromChart` to get an instance of this class.

Historical market depth data is stored at multiple price increments, as determined by the tick size on the chart, for each bar in the chart. The different price increments are identified by their price tick index, where each adjacent price tick index has a price value difference exactly equal to the tick size on the chart.

```
-----*/
```

```
class c_ACSILDepthBars
```

```
{
    friend class c_InternalACSILDepthBars;
```

```
    ///-- Public Methods -----
```

```
public:
```

```
    float GetTickSize();
```

```
    int PriceToTickIndex(float Price);
```

```
    float TickIndexToPrice(int PriceTickIndex);
```

```
    int NumBars();
```

```
    char DepthDataExistsAt(int BarIndex);
```

```
    int GetBarHighestPriceTickIndex(int BarIndex);
```

```
    int GetBarLowestPriceTickIndex(int BarIndex);
```

```
    char GetNextHigherPriceTickIndex(int BarIndex, int& r_PriceTickIndex);
```

```
    BuySellEnum GetMaxDominantSide(int BarIndex, int PriceTickIndex);
```

```
    int GetMaxQuantityForSide(int BarIndex, int PriceTickIndex, BuySellEnum Side);
```

```
    int GetMaxBidQuantity(int BarIndex, int PriceTickIndex);
```

```
    int GetMaxAskQuantity(int BarIndex, int PriceTickIndex);
```

```
    int GetMaxQuantity(int BarIndex, int PriceTickIndex);
```

```
    BuySellEnum GetLastDominantSide(int BarIndex, int PriceTickIndex);
```

```
    int GetLastQuantityForSide(int BarIndex, int PriceTickIndex, BuySellEnum Side);
```

```
    int GetLastBidQuantity(int BarIndex, int PriceTickIndex);
```

```
    int GetLastAskQuantity(int BarIndex, int PriceTickIndex);
```

```
    int GetLastQuantity(int BarIndex, int PriceTickIndex);
```

```
    ///-- Private Types -----
```

```
private:
```

```
    struct s_InternalFunctionPointers;
```

```
    ///-- Private Methods -----
```

```
private:
```

```
    c_ACSILDepthBars
```

```
    ( c_InternalACSILDepthBars* p_Internal
```

```
      , const s_InternalFunctionPointers* p_InternalFunctionPointers
```

```
    );
```

```
    ///-- Private Members -----
```

```
private:
```

```
    c_InternalACSILDepthBars* m_p_Internal;
```

```
    const s_InternalFunctionPointers* m_p_InternalFunctionPointers;
```

//--- Substructures -----

private:

#pragma region s\_InternalFunctionPointers

/\*\*\*\*\*\*

// s\_InternalFunctionPointers

A structure of all of the function pointers used internally by this class.

-----\*/

struct s\_InternalFunctionPointers

{

//--- Members -----

float (SCDLLCALL\* fp\_GetTickSize)

( c\_InternalACSILDepthBars\* p\_Internal  
);

int (SCDLLCALL\* fp\_PriceToTickIndex)

( c\_InternalACSILDepthBars\* p\_Internal  
, float Price  
);

float (SCDLLCALL\* fp\_TickIndexToPrice)

( c\_InternalACSILDepthBars\* p\_Internal  
, int PriceTickIndex  
);

int (SCDLLCALL\* fp\_NumBars)

( c\_InternalACSILDepthBars\* p\_Internal  
);

char (SCDLLCALL\* fp\_DepthDataExstsAt)

( c\_InternalACSILDepthBars\* p\_Internal  
, int BarIndex  
);

int (SCDLLCALL\* fp\_GetBarHighestPriceTickIndex)

( c\_InternalACSILDepthBars\* p\_Internal  
, int BarIndex  
);

int (SCDLLCALL\* fp\_GetBarLowestPriceTickIndex)

( c\_InternalACSILDepthBars\* p\_Internal  
, int BarIndex  
);

char (SCDLLCALL\* fp\_GetNextHigherPriceTickIndex)

( c\_InternalACSILDepthBars\* p\_Internal  
, int BarIndex  
, int& r\_PriceTickIndex  
);

BuySellEnum (SCDLLCALL\* fp\_GetMaxDominantSide)

( c\_InternalACSILDepthBars\* p\_Internal  
, int BarIndex  
, int PriceTickIndex  
);

int (SCDLLCALL\* fp\_GetMaxQuantityForSide)

( c\_InternalACSILDepthBars\* p\_Internal  
, int BarIndex  
, int PriceTickIndex  
, BuySellEnum Side  
);

int (SCDLLCALL\* fp\_GetMaxBidQuantity)

( c\_InternalACSILDepthBars\* p\_Internal  
, int BarIndex  
, int PriceTickIndex  
);

int (SCDLLCALL\* fp\_GetMaxAskQuantity)

( c\_InternalACSILDepthBars\* p\_Internal  
, int BarIndex  
, int PriceTickIndex  
);

```

    int (SCDLLCALL* fp_GetMaxQuantity)
        ( c_InternalACSILDepthBars* p_Internal
        , int BarIndex
        , int PriceTickIndex
        );
    BuySellEnum (SCDLLCALL* fp_GetLastDominantSide)
        ( c_InternalACSILDepthBars* p_Internal
        , int BarIndex
        , int PriceTickIndex
        );
    int (SCDLLCALL* fp_GetLastQuantityForSide)
        ( c_InternalACSILDepthBars* p_Internal
        , int BarIndex
        , int PriceTickIndex
        , BuySellEnum Side
        );
    int (SCDLLCALL* fp_GetLastBidQuantity)
        ( c_InternalACSILDepthBars* p_Internal
        , int BarIndex
        , int PriceTickIndex
        );
    int (SCDLLCALL* fp_GetLastAskQuantity)
        ( c_InternalACSILDepthBars* p_Internal
        , int BarIndex
        , int PriceTickIndex
        );
    int (SCDLLCALL* fp_GetLastQuantity)
        ( c_InternalACSILDepthBars* p_Internal
        , int BarIndex
        , int PriceTickIndex
        );
};

/*****
#pragma endregion
*/

};

/*=====*/
inline float c_ACSILDepthBars::GetTickSize()
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_GetTickSize == NULL
        )
    {
        return 1.0f;
    }

    return m_p_InternalFunctionPointers->fp_GetTickSize(m_p_Internal);
}

/*=====*/
inline int c_ACSILDepthBars::PriceToTickIndex(float Price)
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_PriceToTickIndex == NULL
        )
    {
        return 0;
    }

    return m_p_InternalFunctionPointers->fp_PriceToTickIndex
        ( m_p_Internal
        , Price
        );
}

```

```

/*=====*/
inline float c_ACSILDepthBars::TickIndexToPrice(int PriceTickIndex)
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_TickIndexToPrice == NULL
    )
    {
        return 0.0f;
    }

    return m_p_InternalFunctionPointers->fp_TickIndexToPrice
        ( m_p_Internal
        , PriceTickIndex
        );
}

/*=====*/
inline int c_ACSILDepthBars::NumBars()
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_NumBars == NULL
    )
    {
        return 0;
    }

    return m_p_InternalFunctionPointers->fp_NumBars(m_p_Internal);
}

/*=====*/
inline char c_ACSILDepthBars::DepthDataExistsAt(int BarIndex)
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_DepthDataExstsAt == NULL
    )
    {
        return 0;
    }

    return m_p_InternalFunctionPointers->fp_DepthDataExstsAt
        ( m_p_Internal
        , BarIndex
        );
}

/*=====*/
inline int c_ACSILDepthBars::GetBarHighestPriceTickIndex(int BarIndex)
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_GetBarHighestPriceTickIndex == NULL
    )
    {
        return 0;
    }

    return m_p_InternalFunctionPointers->fp_GetBarHighestPriceTickIndex
        ( m_p_Internal
        , BarIndex
        );
}

/*=====*/
inline int c_ACSILDepthBars::GetBarLowestPriceTickIndex(int BarIndex)
{

```

```

    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_GetBarLowestPriceTickIndex == NULL
    )
    {
        return 0;
    }

    return m_p_InternalFunctionPointers->fp_GetBarLowestPriceTickIndex
        ( m_p_Internal
          , BarIndex
        );
}

/*=====*/
inline char c_ACSILDepthBars::GetNextHigherPriceTickIndex
( int BarIndex
, int& r_PriceTickIndex
)
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_GetNextHigherPriceTickIndex == NULL
    )
    {
        return 0;
    }

    return m_p_InternalFunctionPointers->fp_GetNextHigherPriceTickIndex
        ( m_p_Internal
          , BarIndex
          , r_PriceTickIndex
        );
}

/*=====*/
inline BuySellEnum c_ACSILDepthBars::GetMaxDominantSide
( int BarIndex
, int PriceTickIndex
)
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_GetMaxDominantSide == NULL
    )
    {
        return BSE_UNDEFINED;
    }

    return m_p_InternalFunctionPointers->fp_GetMaxDominantSide
        ( m_p_Internal
          , BarIndex
          , PriceTickIndex
        );
}

/*=====*/
inline int c_ACSILDepthBars::GetMaxQuantityForSide
( int BarIndex
, int PriceTickIndex
, BuySellEnum Side
)
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_GetMaxQuantityForSide == NULL
    )
    {
        return 0;
    }

```

```

}

return m_p_InternalFunctionPointers->fp_GetMaxQuantityForSide
    ( m_p_Internal
      , BarIndex
      , PriceTickIndex
      , Side
    );
}

/*=====*/
inline int c_ACSILDepthBars::GetMaxBidQuantity
( int BarIndex
, int PriceTickIndex
)
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_GetMaxBidQuantity == NULL
    )
    {
        return 0;
    }

    return m_p_InternalFunctionPointers->fp_GetMaxBidQuantity
        ( m_p_Internal
          , BarIndex
          , PriceTickIndex
        );
}

/*=====*/
inline int c_ACSILDepthBars::GetMaxAskQuantity
( int BarIndex
, int PriceTickIndex
)
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_GetMaxAskQuantity == NULL
    )
    {
        return 0;
    }

    return m_p_InternalFunctionPointers->fp_GetMaxAskQuantity
        ( m_p_Internal
          , BarIndex
          , PriceTickIndex
        );
}

/*=====*/
inline int c_ACSILDepthBars::GetMaxQuantity(int BarIndex, int PriceTickIndex)
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_GetMaxQuantity == NULL
    )
    {
        return 0;
    }

    return m_p_InternalFunctionPointers->fp_GetMaxQuantity
        ( m_p_Internal
          , BarIndex
          , PriceTickIndex
        );
}

```

```

}

/*=====*/
inline BuySellEnum c_ACSILDepthBars::GetLastDominantSide
( int BarIndex
, int PriceTickIndex
)
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_GetLastDominantSide == NULL
    )
    {
        return BSE_UNDEFINED;
    }

    return m_p_InternalFunctionPointers->fp_GetLastDominantSide
        ( m_p_Internal
        , BarIndex
        , PriceTickIndex
        );
}

/*=====*/
inline int c_ACSILDepthBars::GetLastQuantityForSide
( int BarIndex
, int PriceTickIndex
, BuySellEnum Side
)
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_GetLastQuantityForSide == NULL
    )
    {
        return 0;
    }

    return m_p_InternalFunctionPointers->fp_GetLastQuantityForSide
        ( m_p_Internal
        , BarIndex
        , PriceTickIndex
        , Side
        );
}

/*=====*/
inline int c_ACSILDepthBars::GetLastBidQuantity
( int BarIndex
, int PriceTickIndex
)
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_GetLastBidQuantity == NULL
    )
    {
        return 0;
    }

    return m_p_InternalFunctionPointers->fp_GetLastBidQuantity
        ( m_p_Internal
        , BarIndex
        , PriceTickIndex
        );
}

/*=====*/

```

```

inline int c_ACSILDepthBars::GetLastAskQuantity
( int BarIndex
, int PriceTickIndex
)
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_GetLastAskQuantity == NULL
    )
    {
        return 0;
    }

    return m_p_InternalFunctionPointers->fp_GetLastAskQuantity
        ( m_p_Internal
        , BarIndex
        , PriceTickIndex
        );
}

/*=====*/
inline int c_ACSILDepthBars::GetLastQuantity(int BarIndex, int PriceTickIndex)
{
    if (m_p_InternalFunctionPointers == NULL
        || m_p_InternalFunctionPointers->fp_GetLastQuantity == NULL
    )
    {
        return 0;
    }

    return m_p_InternalFunctionPointers->fp_GetLastQuantity
        ( m_p_Internal
        , BarIndex
        , PriceTickIndex
        );
}

/*=====*/
inline c_ACSILDepthBars::c_ACSILDepthBars
( c_InternalACSILDepthBars* p_Internal
, const s_InternalFunctionPointers* p_InternalFunctionPointers
)
: m_p_Internal(p_Internal)
, m_p_InternalFunctionPointers(p_InternalFunctionPointers)
{
}

/*=====*/

#endif

```