

**Harbour Guide**

## **hb\_parc()**

Retrieve a string parameter

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_parc( int iParam, ... ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_parclen()**

Retrieve a string parameter length

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_parclen( int iParam, ... ) --> ( ULONG )ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_parcsiz()**

Retrieve a by-reference string parameter length, including terminator

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_parcsiz( int iParam, ... ) --> ( ULONG )ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_pards()**

Retrieve a date as a string `yyyymmdd`

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_pards( int iParam, ... ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is `vm`

### **Platforms**

All

## **hb\_pardsbuff()**

Retrieve a date as a string `yyyymmdd`

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_pardsbuff( char * szDate, int iParam, ... ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is `vm`

### **Platforms**

All

## **hb\_parinfa()**

Retrieve length or element type of an array parameter

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_parinfa( int iParamNum, ULONG uiArrayIndex ) --> ( ULONG )ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_parinfo()**

Determine the param count or data type

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_parinfo( int iParam ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_parl()**

Retrieve a logical parameter as an int

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_parl( int iParam, ... ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_parnd( )**

Retrieve a numeric parameter as a double

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_parnd( int iParam, ... ) --> ( double )dResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_parni()**

Retrieve a numeric parameter as a integer

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_parni( int iParam, ... ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_parnl()**

Retrieve a numeric parameter as a long

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_parnl( int iParam, ... ) --> ( long )lResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_param()**

Retrieve a direct pointer to an item parameter

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_param( int iParam, int iMask ) --> ( PHB_ITEM ) pResult
```

### **Arguments**

**<iParam>**     The 1-based parameter to retrieve.

### **Returns**

**hb\_param()**   returns a direct pointer to an item on the eval stack.

### **Description**

This item will be removed (set to NIL) after a function cleanup, so if the item needs to survive the current function (e.g. copied to a static) you should use `hb_itemParam` instead.

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

### **See Also:**

[hb\\_itemParam\(\)](#)

## **hb\_pcount()**

Returns the number of supplied parameters

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_pcount( void ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

Note that when HB\_API\_MACROS is defined, this function is replaced with a macro: `hb_pcount() --> ( ( int ) hb_stack.pBase->item.asSymbol.paramcnt )`

### **Examples**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_ret()**

Post a NIL return value

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_ret( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

Note that when HB\_API\_MACROS is defined, this function is replaced with a macro: `hb_ret() --> hb_itemClear( &hb_stack.Return )`

### **Examples**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_retc()**

Returns a string

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_retc( char * szText ) --> void
```

### **Arguments**

### **Returns**

### **Description**

Note that when HB\_API\_MACROS is defined, this function is replaced with a macro: `hb_retc( szText ) --> hb_itemPutC( &hb_stack.Return, szText )`

### **Examples**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_retclen()**

Returns a string with a specific length

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_retclen( char * szText, ULONG ulLen ) --> void
```

### **Arguments**

### **Returns**

### **Description**

Note that when HB\_API\_MACROS is defined, this function is replaced with a macro: hb\_retclen( szText, ulLen ) --> hb\_itemPutCL( &hb\_stack.Return, szText, ulLen )

### **Examples**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_retds()**

Returns a date, must use yyyymmdd format

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_retds( char * szDate ) --> void
```

### **Arguments**

### **Returns**

### **Description**

Note that when HB\_API\_MACROS is defined, this function is replaced with a macro: `hb_retds( szDate ) --> hb_itemPutDS( &hb_stack.Return, szDate )`

### **Examples**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

**hb\_retd()**  
Returns a date

## Syntax

C Prototype

```
#include <hbapi.h>
hb_retd( long lYear, long lMonth, long lDay ) --> void
```

## Arguments

## Returns

## Description

Note that when HB\_API\_MACROS is defined, this function is replaced with a macro: `hb_retd( lYear, lMonth, lDay ) --> hb_itemPutD( &hb_stack.Return, lYear, lMonth, lDay )`

## Examples

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

## **hb\_retdl()**

Returns a long value as a julian date

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_retdl( long lJulian ) --> void
```

### **Arguments**

### **Returns**

### **Description**

Note that when HB\_API\_MACROS is defined, this function is replaced with a macro: `hb_retdl( lJulian ) --> hb_itemPutDL( &hb_stack.Return, lJulian )`

### **Examples**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_retl()**

Returns a logical integer

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_retl( int iTrueFalse ) --> void
```

### **Arguments**

### **Returns**

### **Description**

Note that when HB\_API\_MACROS is defined, this function is replaced with a macro: `hb_retl( iLogical ) --> hb_itemPutL( &hb_stack.Return, iLogical ? TRUE : FALSE )`

### **Examples**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_retn()**

Returns a double

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_retn( double dNumber ) --> void
```

### **Arguments**

### **Returns**

### **Description**

Note that when HB\_API\_MACROS is defined, this function is replaced with a macro: `hb_retn( dNumber ) --> hb_itemPutND( &hb_stack.Return, dNumber )`

### **Examples**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_retni()**

Returns a integer number

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_retni( int iNumber ) --> void
```

### **Arguments**

### **Returns**

### **Description**

Note that when HB\_API\_MACROS is defined, this function is replaced with a macro: `hb_retni( iNumber ) --> hb_itemPutNI( &hb_stack.Return, iNumber )`

### **Examples**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_retnl()**

Returns a long number

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_retnl( long lNumber ) --> void
```

### **Arguments**

### **Returns**

### **Description**

Note that when HB\_API\_MACROS is defined, this function is replaced with a macro: `hb_retnl( lNumber ) --> hb_itemPutNL( &hb_stack.Return, lNumber )`

### **Examples**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_retnlen()**

Returns a double, with specific width and decimals

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_retnlen( double dNumber, int iWidth, int iDec ) --> void
```

### **Arguments**

### **Returns**

### **Description**

Note that when HB\_API\_MACROS is defined, this function is replaced with a macro: `hb_retnlen( dNumber, iWidth, iDec ) --> hb_itemPutNLen( &hb_stack.Return, dNumber, iWidth, iDec )`

### **Examples**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_retnklen( )**

Returns a double, with specific width and decimals

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_retnklen( double dNumber, int iWidth, int iDec ) --> void
```

### **Arguments**

### **Returns**

### **Description**

Note that when HB\_API\_MACROS is defined, this function is replaced with a macro: hb\_retnklen( dNumber, iWidth, iDec ) --> hb\_itemPutNDLen( &hb\_stack.Return, dNumber, iWidth, iDec )

### **Examples**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_retnilen()**

Returns a integer number, with specific width

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_retnilen( int iNumber, int iWidth ) --> void
```

### **Arguments**

### **Returns**

### **Description**

Note that when HB\_API\_MACROS is defined, this function is replaced with a macro: hb\_retnilen( iNumber, iWidth ) --> hb\_itemPutNILen( &hb\_stack.Return, iNumber, iWidth )

### **Examples**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_retnllen()**

Returns a long number, with specific width

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_retnllen( long lNumber, int iWidth ) --> void
```

### **Arguments**

### **Returns**

### **Description**

Note that when HB\_API\_MACROS is defined, this function is replaced with a macro: hb\_retnllen( lNumber, iWidth ) --> hb\_itemPutNLLen( &hb\_stack.Return, lNumber, iWidth )

### **Examples**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_reta()**

Returns an array with a specific length

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_reta( ULONG ulLen ) --> void
```

### **Arguments**

### **Returns**

### **Description**

Note that when HB\_API\_MACROS is defined, this function is replaced with a macro: `hb_reta( ulLen ) --> hb_arrayNew( &hb_stack.Return, ulLen )`

### **Examples**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_storc()**

Stores a szString on a variable by reference

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_storc( char * szText, int iParam, ... ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_storclen()**

Stores a fixed length string on a variable by reference

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_storclen( char * szText, ULONG ulLength, int iParam, ... ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_stords()**

SzDate must have yyyyymmdd format

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_stords( char * szDate, int iParam, ... ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_storl()**

Stores a logical integer on a variable by reference

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_storl( int iLogical, int iParam, ... ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_storni()**

Stores an integer on a variable by reference

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_storni( int iValue, int iParam, ... ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_stornl()**

Stores a long on a variable by reference

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_stornl( long lValue, int iParam, ... ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_stornd()**

Stores a double on a variable by reference

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_stornd( double dValue, int iParam, ... ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_xinit()**

Initialize fixed memory subsystem

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_xinit( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_xexit()**

Deinitialize fixed memory subsystem

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_xexit( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_xalloc()**

Allocates memory, returns NULL on failure

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_xalloc( ULONG ulSize ) --> ( void * )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_xgrab( )**

Allocates memory, exits on failure

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_xgrab( ULONG ulSize ) --> ( void * )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_xfree()**

Frees memory

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_xfree( void * pMem ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

**hb\_xrealloc()**  
Reallocates memory

## Syntax

C Prototype

```
#include <hbapi.h>
hb_xrealloc( void * pMem, ULONG ulSize ) --> ( void * )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

## **hb\_xsize()**

Returns the size of an allocated memory block

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_xsize( void * pMem ) --> ( ULONG )ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_xquery()**

Query different types of memory information

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_xquery( USHORT uiMode ) --> ( ULONG )ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_xmemcpy( )**

Copy more than memcpy() can

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_xmemcpy( void * pDestArg, void * pSourceArg, ULONG ulLen ) --> ( void * )pResult
```

### **Arguments**

### **Returns**

### **Description**

If UINT\_MAX is defined as ULONG\_MAX then this function is replaced by a macro replacement to memcpy()

### **Examples**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_xmemset()**

Set more than memset() can

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_xmemset( void * pDestArg, int iFill, ULONG ulLen ) --> ( void * )pResult
```

### **Arguments**

### **Returns**

### **Description**

If UINT\_MAX is defined as ULONG\_MAX then this function is replaced by a macro replacement to memset()

### **Examples**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

**hb\_arrayNew( )**  
Creates a new array

## Syntax

C Prototype

```
#include <hbapi.h>
hb_arrayNew( PHB_ITEM pItem, ULONG ulLen ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

## **hb\_arrayLen()**

Retrives the array len

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayLen( PHB_ITEM pArray ) --> ( ULONG )ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayIsObject()**

Retrives if the array is an object

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayIsObject( PHB_ITEM pArray ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayAdd()**

Add a new item to the end of an array item

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayAdd( PHB_ITEM pArray, PHB_ITEM pItemValue ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayIns()**

Insert a nil item into an array, without changing the length

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayIns( PHB_ITEM pArray, ULONG ulIndex ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayDel()**

Delete an array item, without changing length

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_arrayDel( PHB_ITEM pArray, ULONG ulIndex ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arraySize()**

Sets the array total length

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arraySize( PHB_ITEM pArray, ULONG ulLen ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayLast()**

Retrieve last item in an array

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayLast( PHB_ITEM pArray, PHB_ITEM pResult ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayRelease()**

Releases an array - don't call it - use ItemRelease() !!!

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayRelease( PHB_ITEM pArray ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

**hb\_arraySet()**  
Sets an array element

## Syntax

C Prototype

```
#include <hbapi.h>
hb_arraySet( PHB_ITEM pArray, ULONG ulIndex, PHB_ITEM pItem ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

**hb\_arrayGet()**  
Retrieves an item

## Syntax

C Prototype

```
#include <hbapi.h>
hb_arrayGet( PHB_ITEM pArray, ULONG ulIndex, PHB_ITEM pItem ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

## **hb\_arrayGetItemPtr()**

Returns pointer to specified element of the array

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayGetItemPtr( PHB_ITEM pArray, ULONG ulIndex ) --> ( PHB_ITEM )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayCopyC()**

Copy a string into an array item

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_arrayCopyC( PHB_ITEM pArray, ULONG ulIndex, char * szBuffer, ULONG ulLen ) --> (
ULONG )ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayGetC()**

Retrieves the string contained on an array element

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayGetC( PHB_ITEM pArray, ULONG ulIndex ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayGetCPtr()**

Retrieves the string pointer on an array element

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayGetCPtr( PHB_ITEM pArray, ULONG ulIndex ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayGetCLen( )**

Retrieves the string length contained on an array element

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayGetCLen( PHB_ITEM pArray, ULONG ulIndex ) --> ( ULONG )ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayGetL()**

Retrieves the logical value contained on an array element

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayGetL( PHB_ITEM pArray, ULONG ulIndex ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayGetNI()**

Retrieves the int value contained on an array element

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayGetNI( PHB_ITEM pArray, ULONG ulIndex ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayGetNL()**

Retrieves the long numeric value contained on an array element

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayGetNL( PHB_ITEM pArray, ULONG ulIndex ) --> ( long )lResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayGetND( )**

Retrieves the double value contained on an array element

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayGetND( PHB_ITEM pArray, ULONG ulIndex ) --> ( double )dResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayGetDS()**

Retrieves the date value contained in an array element

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_arrayGetDS( PHB_ITEM pArray, ULONG ulIndex, char * szDate ) --> ( char *
)pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayGetDL( )**

Retrieves the date value contained in an array element, as a long integer

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayGetDL( PHB_ITEM pArray, ULONG ulIndex ) --> ( long )lResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayGetType( )**

Retrieves the type of an array item

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayGetType( PHB_ITEM pArray, ULONG ulIndex ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayFill()**

Fill an array with a given item

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_arrayFill( PHB_ITEM pArray, PHB_ITEM pValue, ULONG * pulStart, ULONG * pulCount )
--> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayScan()**

Scan an array for a given item, or until code-block item returns TRUE

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_arrayScan( PHB_ITEM pArray, PHB_ITEM pValue, ULONG * pulStart, ULONG * pulCount )
--> ( ULONG )ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayEval()**

Execute a code-block for every element of an array item

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_arrayEval( PHB_ITEM pArray, PHB_ITEM bBlock, ULONG * pulStart, ULONG * pulCount )
--> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayCopy()**

Copy items from one array to another

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_arrayCopy( PHB_ITEM pSrcArray, PHB_ITEM pDstArray, ULONG * pulStart, ULONG *
pulCount, ULONG * pulTarget ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayClone()**

Returns a duplicate of an existing array, including all nested items

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_arrayClone( PHB_ITEM pArray ) --> ( PHB_ITEM )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arraySort()**

Sorts an array item

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_arraySort( PHB_ITEM pArray, ULONG * pulStart, ULONG * pulCount, PHB_ITEM pBlock )
--> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_stricmp()**

Compare two strings without regards to case

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_stricmp( const char * s1, const char * s2 ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_strnicmp()**

Compare two string without regards to case, limited by length

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_strnicmp( const char * s1, const char * s2, ULONG ulLen ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_strupr()**

Convert a string in-place to upper-case

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_strupr( char * pszText ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_strdup()**

Returns a pointer to a newly allocated copy of the source string

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_strdup( const char * pszText ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_strMatchRegExp( )**

Compare two strings using a regular expression pattern

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_strMatchRegExp( const char * szString, const char * szMask ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_strEmpty()**

Returns whether a string contains only white space

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_strEmpty( const char * szText, ULONG ulLen ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_strDescend( )**

Copy a string to a buffer, inverting each character

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_strDescend( char * szStringTo, const char * szStringFrom, ULONG ulLen ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_strAt()**

Returns an index to a sub-string within another string

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_strAt( const char * szSub, ULONG ulSubLen, const char * szText, ULONG ulLen ) -->
( ULONG )ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_strUpper()**

Convert an existing string buffer to upper case

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_strUpper( char * szText, ULONG ulLen ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_strLower()**

Convert an existing string buffer to lower case

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_strLower( char * szText, ULONG ulLen ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_strncpyUpper()**

Copy an existing string buffer to another buffer, as upper case

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_strncpyUpper( char * pDest, const char * pSource, ULONG ulLen ) --> ( char *
)pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_strVal()**

Return the numeric value of a character string representation of a number

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_strVal( const char * szText, ULONG ulLen ) --> ( double )dResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_strLTrim()**

Return a pointer to the first non-white space character

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_strLTrim( const char * szText, ULONG * ulLen ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_strRTrimLen()**

Return length of a string, ignoring trailing white space (or true spaces)

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_strRTrimLen( const char * szText, ULONG ulLen, BOOL bAnySpace ) --> ( ULONG
)ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_numRound( )**

Round a number to a specific number of digits

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_numRound( double dResult, int iDec ) --> ( double )dResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_clsReleaseAll()**

Releases all defined classes

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_clsReleaseAll( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_objGetClsName()**

Retrieves an object class name

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_objGetClsName( PHB_ITEM pObject ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_objGetMethod()**

Returns the method pointer of a object class

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_objGetMethod( PHB_ITEM pObject, PHB_SYMB pSymMsg ) --> ( PHB_FUNC )hResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_objHasMsg()**

Returns TRUE/FALSE whether szString is an existing message for object

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_objHasMsg( PHB_ITEM pObject, char * szString ) --> ( ULONG )ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_dynsymGet()**

Finds and creates a dynamic symbol if not found

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_dynsymGet( char * szName ) --> ( PHB_DYNS )hResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_dynsymNew( )**

Creates a new dynamic symbol based on a local one

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_dynsymNew( PHB_SYMB pSymbol ) --> ( PHB_DYNS )hResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_dynsymFind()**

Finds a dynamic symbol

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_dynsymFind( char * szName ) --> ( PHB_DYNS )hResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_dynsymFindName( )**

Converts to uppercase and finds a dynamic symbol

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_dynsymFindName( char * szName ) --> ( PHB_DYNS )hResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_dynsymLog( )**

Displays all dynamic symbols

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_dynsymLog( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_dynsymRelease()**

Releases the memory of the dynamic symbol table

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_dynsymRelease( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_dynsymEval()**

Enumerates all dynamic symbols

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_dynsymEval( PHB_DYNS_FUNC pFunction, void * Cargo ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_cmdargInit()**

Initialize command line argument API's

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_cmdargInit( int argc, char * argv[] ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_cmdargARGC( )**

Retrieve command line argument count

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_cmdargARGC( void ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_cmdargARGV( )**

Retrieve command line argument buffer pointer

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_cmdargARGV( void ) --> ( char ** )ppszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_cmdargIsInternal()**

Determine if a string is an internal setting

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_cmdargIsInternal( const char * szArg ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_cmdargCheck()**

Check if a given internal switch (like //INFO) was set

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_cmdargCheck( const char * pszName ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_cmdargString()**

Returns the string value of an internal switch (like //TEMPPATH:"C:\")

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_cmdargString( const char * pszName ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_cmdargNum( )**

Returns the numeric value of an internal switch (like //F:90)

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_cmdargNum( const char * pszName ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_cmdargProcessVM( )**

Check for command line internal arguments

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_cmdargProcessVM( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_symbolNew( )**

Create a new symbol

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_symbolNew( char * szName ) --> ( PHB_SYMB )hResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_codeblockNew( )**

Create a code-block

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_codeblockNew( BYTE * pBuffer, USHORT uiLocals, USHORT * pLocalPosTable, PHB_SYMB
pSymbols ) --> ( HB_CODEBLOCK_PTR )hResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## hb\_codeblockMacroNew( )

### Syntax

C Prototype

```
#include <hbapi.h>
hb_codeblockMacroNew( BYTE * pBuffer, USHORT usLen ) --> ( HB_CODEBLOCK_PTR )hResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is vm

### Platforms

All

## **hb\_codeblockDelete()**

Delete a codeblock

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_codeblockDelete( HB_ITEM_PTR pItem ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_codeblockGetVar()**

Get local variable referenced in a codeblock

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_codeblockGetVar( PHB_ITEM pItem, LONG iItemPos ) --> ( PHB_ITEM )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_codeblockGetRef()**

Get local variable passed by reference

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_codeblockGetRef( PHB_ITEM pItem, PHB_ITEM pRefer ) --> ( PHB_ITEM )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_codeblockEvaluate()**

Evaluate a codeblock

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_codeblockEvaluate( HB_ITEM_PTR pItem ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

**hb\_codeblockCopy()**  
Copy a codeblock

## Syntax

C Prototype

```
#include <hbapi.h>
hb_codeblockCopy( PHB_ITEM pDest, PHB_ITEM pSource ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

## **hb\_memvarValueNew( )**

Create a new global value

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_memvarValueNew( HB_ITEM_PTR pSource, BOOL bTrueMemvar ) --> ( HB_HANDLE )hResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_memvarValueBaseAddress()**

Retrieve the base address of the values table

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_memvarValueBaseAddress( void ) --> ( HB_VALUE_PTR * )phResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_memvarsInit()**

Initialize the memvar API system

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_memvarsInit( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_memvarsRelease()**

Clear all PUBLIC and PRIVATE variables

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_memvarsRelease( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_memvarsFree()**

Release the memvar API system

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_memvarsFree( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_memvarValueIncRef()**

Increase the reference count of a global value

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_memvarValueIncRef( HB_HANDLE hValue ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_memvarValueDecRef()**

Decrease the reference count of a global value

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_memvarValueDecRef( HB_HANDLE hValue ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_memvarSetValue()**

Copy an item into a symbol

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_memvarSetValue( PHB_SYMB pMemvarSymb, HB_ITEM_PTR pItem ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_memvarGet()**

Copy an symbol value into an item

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_memvarGet( HB_ITEM_PTR pItem, PHB_SYMB pMemvarSymb ) --> ( ERRCODE )hResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_memvarGetValue()**

Copy an symbol value into an item, with error trapping

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_memvarGetValue( HB_ITEM_PTR pItem, PHB_SYMB pMemvarSymb ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_memvarGetRefer()**

Copy a reference to a symbol value into an item, with error trapping

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_memvarGetRefer( HB_ITEM_PTR pItem, PHB_SYMB pMemvarSymb ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_memvarGetPrivatesBase()**

Retrieve current PRIVATE variables stack base

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_memvarGetPrivatesBase( void ) --> ( ULONG )ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_memvarSetPrivatesBase()**

Release PRIVATE variables created after specified base

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_memvarSetPrivatesBase( ULONG ulBase ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

**hb\_memvarNewParameter( )**

## Syntax

C Prototype

```
#include <hbapi.h>
hb_memvarNewParameter( PHB_SYMB pSymbol, PHB_ITEM pValue ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

## **hb\_memvarGetStrValuePtr()**

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_memvarGetStrValuePtr( char * szVarName, ULONG *pulLen ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_memvarCreateFromItem( )**

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_memvarCreateFromItem( PHB_ITEM pMemvar, BYTE bScope, PHB_ITEM pValue ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_memvarScope()**

Retrieve scope of a dynamic variable symbol

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_memvarScope( char * szVarName, ULONG ulLength ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_conInit()**

Initialize the console API system

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_conInit( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_conRelease()**

Release the console API system

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_conRelease( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_conNewLine()**

Retrieve a pointer to a static buffer containing new-line characters

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_conNewLine( void ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_conOutStd()**

Output an string to STDOUT

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_conOutStd( char * pStr, ULONG ulLen ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_conOutErr()**

Output an string to STDERR

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_conOutErr( char * pStr, ULONG ulLen ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_conSetCursor()**

Retrieve and optionally set cursor shape

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_conSetCursor( BOOL bSetCursor, USHORT usNewCursor ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_conSetColor()**

Retrieve and optionally set console color

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_conSetColor( char * szColor ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_conXSaveRestRelease()**

Release the save/restore API

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_conXSaveRestRelease( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_compReservedName( )**

Determines if a string contains a reserve word

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_compReservedName( char * szName ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_procname()**

Retrieve a procedure name into a buffer

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_procname( int iLevel, char * szName ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_macroGetValue()**

Retrieve results of a macro expansion

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_macroGetValue( HB_ITEM_PTR pItem ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_macroSetValue()**

Assign a value to a macro-expression item

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_macroSetValue( HB_ITEM_PTR pItem ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

**hb\_macroTextValue()**  
Macro text substitution

## Syntax

C Prototype

```
#include <hbapi.h>
hb_macroTextValue( HB_ITEM_PTR pItem ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

## **hb\_macroPushSymbol()**

Handle a macro function calls, e.g. `var := &macro()`

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_macroPushSymbol( HB_ITEM_PTR pItem ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_macroRun()**

Executes pcode compiled by macro compiler

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_macroRun( HB_MACRO_PTR pMacro ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_macroCompile()**

Compile a string and return a pcode buffer

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_macroCompile( char * szString ) --> ( HB_MACRO_PTR )hResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_macroDelete()**

Release all memory allocated for macro evaluation

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_macroDelete( HB_MACRO_PTR pMacro ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_macroTextSubst()**

Substitute macro variables occurrences within a given string

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_macroTextSubst( char * szString, ULONG *pulStringLen ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_macroIsIdent()**

Determine if a string is a valid function or variable name

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_macroIsIdent( char * szString ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_macroPopAliasedValue()**

Compiles and evaluates an aliased macro expression

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_macroPopAliasedValue( HB_ITEM_PTR pAlias, HB_ITEM_PTR pVar ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_macroPushAliasedValue()**

Compiles and evaluates an aliased macro expression

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_macroPushAliasedValue( HB_ITEM_PTR pAlias, HB_ITEM_PTR pVar ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_macroGetType()**

Determine the type of an expression

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_macroGetType( HB_ITEM_PTR pItem ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_gcAlloc()**

Allocates a memory controlled by the garbage collector

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_gcAlloc( ULONG ulSize, HB_GARBAGE_FUNC_PTR pFunc ) --> ( void * )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_gcFree()**

Deallocates a memory allocated by the garbage collector

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_gcFree( void *pAlloc ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_gcLock( )**

Do not release passed memory block

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_gcLock( void *pAlloc ) --> ( void * )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_gcUnlock()**

Passed block is allowed to be released

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_gcUnlock( void *pAlloc ) --> ( void * )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_gcLockItem()**

Do not release a memory block stored inside an item

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_gcLockItem( HB_ITEM_PTR pItem ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_gcUnlockItem( )**

Allow to release the item

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_gcUnlockItem( HB_ITEM_PTR pItem ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_gcCollect()**

Checks if a single memory block can be released

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_gcCollect( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_gcCollectAll()**

Checks if all memory blocks can be released

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_gcCollectAll( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_gcItemRef()**

Checks if passed item refers passed memory block pointer

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_gcItemRef( HB_ITEM_PTR pItem ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmIsLocalRef()**

Hvm.c - mark all local variables as used

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_vmIsLocalRef( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmIsStaticRef()**

Hvm.c - mark all static variables as used

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_vmIsStaticRef( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

**hb\_memvarsIsMemvarRef( )**  
Memvars.c - mark all memvar variables as used

## Syntax

C Prototype

```
#include <hbapi.h>
hb_memvarsIsMemvarRef( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

## **hb\_clsIsClassRef()**

Classes.c - mark all class internals as used

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_clsIsClassRef( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_codeblockDeleteGarbage()**

Clear a codeblock before releasing by the GC

### **Syntax**

C Prototype

```
#include <hbapi.h>
```

```
hb_codeblockDeleteGarbage( void * Cargo ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_arrayReleaseGarbage()**

Clear an array before releasing by the GC

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_arrayReleaseGarbage( void * Cargo ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_idleState()**

Services a single idle state

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_idleState( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_idleReset()**

Services a single idle state

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_idleReset( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_idleShutDown( )**

Closes all background tasks

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_idleShutDown( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_verPlatform()**

Retrieves a newly allocated buffer containing platform version

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_verPlatform( void ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_verCompiler()**

Retrieves a newly allocated buffer containing compiler version

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_verCompiler( void ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_verHarbour( )**

Retrieves a newly allocated buffer containing harbour version

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_verHarbour( void ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_verBuildInfo()**

Display harbour, compiler, and platform versions to standard console

### **Syntax**

C Prototype

```
#include <hbapi.h>
hb_verBuildInfo( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

**HB\_IS\_OF\_TYPE( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
HB_IS_OF_TYPE( p, t ) --> <see ( ( ( p )->type & ~HB_IT_BYREF ) == t )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapi.h

## Platforms

All

**HB\_IS\_BYREF( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
HB_IS_BYREF( p ) --> <see ( ( p )->type & HB_IT_BYREF )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapi.h

## Platforms

All

**HB\_IS\_ARRAY( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
HB_IS_ARRAY( p ) --> <see HB_IS_OF_TYPE( p, HB_IT_ARRAY )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapi.h

## Platforms

All

**HB\_IS\_NIL( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
HB_IS_NIL( p ) --> <see HB_IS_OF_TYPE( p, HB_IT_NIL )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapi.h

## Platforms

All

**HB\_IS\_BLOCK( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
HB_IS_BLOCK( p ) --> <see HB_IS_OF_TYPE( p, HB_IT_BLOCK )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapi.h

## Platforms

All

**HB\_IS\_DATE( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
HB_IS_DATE( p ) --> <see HB_IS_OF_TYPE( p, HB_IT_DATE )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapi.h

## Platforms

All

## HB\_IS\_DOUBLE( )

### Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
HB_IS_DOUBLE( p ) --> <see HB_IS_OF_TYPE( p, HB_IT_DOUBLE )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapi.h

### Platforms

All

**HB\_IS\_INTEGER( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
HB_IS_INTEGER( p ) --> <see HB_IS_OF_TYPE( p, HB_IT_INTEGER )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapi.h

## Platforms

All

## HB\_IS\_LOGICAL( )

### Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
HB_IS_LOGICAL( p ) --> <see HB_IS_OF_TYPE( p, HB_IT_LOGICAL )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapi.h

### Platforms

All

**HB\_IS\_LONG( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
HB_IS_LONG( p ) --> <see HB_IS_OF_TYPE( p, HB_IT_LONG )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapi.h

## Platforms

All

**HB\_IS\_NUMERIC( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
HB_IS_NUMERIC( p ) --> <see ( ( p )->type & HB_IT_NUMERIC )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapi.h

## Platforms

All

## HB\_IS\_OBJECT( )

### Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
HB_IS_OBJECT( p ) --> <see HB_IS_OF_TYPE( p, HB_IT_OBJECT )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapi.h

### Platforms

All

## HB\_IS\_STRING()

### Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
HB_IS_STRING( p ) --> <see ( ( ( p )->type & ~( HB_IT_BYREF | HB_IT_MEMOFLAG ) ) ==
HB_IT_STRING )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapi.h

### Platforms

All

**HB\_IS\_MEMO( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
HB_IS_MEMO( p ) --> <see HB_IS_OF_TYPE( p, HB_IT_MEMO )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapi.h

## Platforms

All

**HB\_IS\_SYMBOL( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
HB_IS_SYMBOL( p ) --> <see HB_IS_OF_TYPE( p, HB_IT_SYMBOL )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapi.h

## Platforms

All

**HB\_IS\_MEMVAR( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
HB_IS_MEMVAR( p ) --> <see HB_IS_OF_TYPE( p, HB_IT_MEMVAR )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapi.h

## Platforms

All

## HB\_IS\_POINTER( )

### Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
HB_IS_POINTER( p ) --> <see HB_IS_OF_TYPE( p, HB_IT_POINTER )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapi.h

### Platforms

All

## ISNIL()

NOTE: Intentionally using a different method

### Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
ISNIL( n ) --> <see ( hb_param( n, HB_IT_ANY ) == NULL || HB_IS_NIL( hb_param( n,
HB_IT_ANY ) ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapi.h

### Platforms

All

## ISCHAR( )

### Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
ISCHAR( n ) --> <see ( hb_param( n, HB_IT_STRING ) != NULL )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapi.h

### Platforms

All

## ISNUM( )

### Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
ISNUM( n ) --> <see ( hb_param( n, HB_IT_NUMERIC ) != NULL )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapi.h

### Platforms

All

## ISLOG( )

### Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
ISLOG( n ) --> <see ( hb_param( n, HB_IT_LOGICAL ) != NULL )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapi.h

### Platforms

All

## ISDATE( )

### Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
ISDATE( n ) --> <see ( hb_param( n, HB_IT_DATE ) != NULL )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapi.h

### Platforms

All

## ISMEMO( )

### Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
ISMEMO( n ) --> <see ( hb_param( n, HB_IT_MEMO ) != NULL )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapi.h

### Platforms

All

## **ISBYREF( )**

NOTE: Intentionally using a different method

### **Syntax**

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
ISBYREF( n ) --> <see ( hb_parinfo( n ) & HB_IT_BYREF )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapi.h

### **Platforms**

All

# ISARRAY( )

## Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
ISARRAY( n ) --> <see ( hb_param( n, HB_IT_ARRAY ) != NULL )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapi.h

## Platforms

All

# ISOBJECT( )

## Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
ISOBJECT( n ) --> <see ( ISARRAY( n ) && hb_param( n, HB_IT_ARRAY
)->asArray.value->uiClass != 0 )>
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Header file is hbapi.h

## Platforms

All

## **ISBLOCK( )**

Not available in CA-Cl\*pper.

### **Syntax**

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
ISBLOCK( n ) --> <see ( hb_param( n, HB_IT_BLOCK ) != NULL )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapi.h

### **Platforms**

All

## **ISPOINTER( )**

Not available in CA-Cl\*pper.

### **Syntax**

C Prototype (macro definition)

```
#include <hbapi.h>
```

```
ISPOINTER( n ) --> <see ( hb_param( n, HB_IT_POINTER ) != NULL )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapi.h

### **Platforms**

All

## HB\_ISSPACE( )

### Syntax

C Prototype (macro definition)

```
#include <hbapi.h>
HB_ISSPACE( c ) --> <see ( ( c ) == ' ' || ( c ) == HB_CHAR_HT || ( c ) ==
HB_CHAR_LF || ( c ) == HB_CHAR_CR )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapi.h

### Platforms

All

# hb\_errGetDescription()

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errGetDescription( PHB_ITEM pError ) --> ( char * )pszResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_errGetFileName( )**

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errGetFileName( PHB_ITEM pError ) --> ( char * )pszResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_errGetFlags()**

### **Syntax**

C Prototype

```
#include <hbapierr.h>
hb_errGetFlags( PHB_ITEM pError ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_errGetGenCode( )**

### **Syntax**

C Prototype

```
#include <hbapierr.h>
hb_errGetGenCode( PHB_ITEM pError ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_errGetOperation()**

### **Syntax**

C Prototype

```
#include <hbapierr.h>
hb_errGetOperation( PHB_ITEM pError ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

**hb\_errGetOsCode( )**

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errGetOsCode( PHB_ITEM pError ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_errGetSeverity()**

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errGetSeverity( PHB_ITEM pError ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_errGetSubCode( )**

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errGetSubCode( PHB_ITEM pError ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_errGetSubSystem()**

### **Syntax**

C Prototype

```
#include <hbapierr.h>
hb_errGetSubSystem( PHB_ITEM pError ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_errGetTries()**

### **Syntax**

C Prototype

```
#include <hbapierr.h>
hb_errGetTries( PHB_ITEM pError ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_errLaunch( )**

### **Syntax**

C Prototype

```
#include <hbapierr.h>
hb_errLaunch( PHB_ITEM pError ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

**hb\_errNew( )**

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errNew( void ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

# hb\_errPutArgs( )

## Syntax

C Prototype

```
#include <hbapierr.h>
```

```
hb_errPutArgs( PHB_ITEM pError, USHORT uiArgCount, ... ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

# hb\_errPutDescription()

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errPutDescription( PHB_ITEM pError, char * szDescription ) --> ( PHB_ITEM
)pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_errPutFileName()**

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errPutFileName( PHB_ITEM pError, char * szFileName ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

# hb\_errPutFlags()

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errPutFlags( PHB_ITEM pError, USHORT uiFlags ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_errPutGenCode()**

### **Syntax**

C Prototype

```
#include <hbapierr.h>
hb_errPutGenCode( PHB_ITEM pError, USHORT uiGenCode ) --> ( PHB_ITEM )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_errPutOperation()**

### **Syntax**

C Prototype

```
#include <hbapierr.h>
hb_errPutOperation( PHB_ITEM pError, char * szOperation ) --> ( PHB_ITEM )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

**hb\_errPutOsCode( )**

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errPutOsCode( PHB_ITEM pError, USHORT uiOsCode ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_errPutSeverity()**

### **Syntax**

C Prototype

```
#include <hbapierr.h>
hb_errPutSeverity( PHB_ITEM pError, USHORT uiSeverity ) --> ( PHB_ITEM )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_errPutSubCode()**

### **Syntax**

C Prototype

```
#include <hbapierr.h>
hb_errPutSubCode( PHB_ITEM pError, USHORT uiSubCode ) --> ( PHB_ITEM )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_errPutSubSystem( )**

### **Syntax**

C Prototype

```
#include <hbapierr.h>
hb_errPutSubSystem( PHB_ITEM pError, char * szSubSystem ) --> ( PHB_ITEM )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

# hb\_errPutTries()

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errPutTries( PHB_ITEM pError, USHORT uiTries ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_errRelease()**

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errRelease( PHB_ITEM pError ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_errInit()**

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errInit( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_errExit()**

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errExit( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_errLaunchSubst( )**

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errLaunchSubst( PHB_ITEM pError ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## hb\_errRT\_New( )

### Syntax

C Prototype

```
#include <hbapierr.h>
hb_errRT_New( USHORT uiSeverity, char * szSubSystem, ULONG ulGenCode, ULONG
ulSubCode, char * szDescription, char * szOperation, USHORT uiOsCode, USHORT
uiFlags ) --> ( PHB_ITEM )pResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is rtl

### Platforms

All

## hb\_errRT\_New\_Subst()

### Syntax

C Prototype

```
#include <hbapierr.h>
hb_errRT_New_Subst( USHORT uiSeverity, char * szSubSystem, ULONG ulGenCode, ULONG
ulSubCode, char * szDescription, char * szOperation, USHORT uiOsCode, USHORT
uiFlags ) --> ( PHB_ITEM )pResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is rtl

### Platforms

All

## **hb\_errrRT\_BASE( )**

### **Syntax**

C Prototype

```
#include <hbapierr.h>
hb_errrRT_BASE( ULONG ulGenCode, ULONG ulSubCode, char * szDescription, char *
szOperation ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

**hb\_errRT\_BASE\_Ext1()**

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errRT_BASE_Ext1( ULONG ulGenCode, ULONG ulSubCode, char * szDescription, char *
szOperation, USHORT uiOsCode, USHORT uiFlags ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_errRT\_BASE\_Subst()**

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errRT_BASE_Subst( ULONG ulGenCode, ULONG ulSubCode, char * szDescription, char *
szOperation ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_errRT\_BASE\_SubstR( )**

### **Syntax**

C Prototype

```
#include <hbapierr.h>
hb_errRT_BASE_SubstR( ULONG ulGenCode, ULONG ulSubCode, char * szDescription, char *
szOperation ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

**hb\_errRT\_TERM( )**

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errRT_TERM( ULONG ulGenCode, ULONG ulSubCode, char * szDescription, char *
szOperation, USHORT uiOSCode, USHORT uiFlags ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_errRT\_DBCMD( )**

### **Syntax**

C Prototype

```
#include <hbapierr.h>
hb_errRT_DBCMD( ULONG ulGenCode, ULONG ulSubCode, char * szDescription, char *
szOperation ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_errRT\_TOOLS( )**

### **Syntax**

C Prototype

```
#include <hbapierr.h>
hb_errRT_TOOLS( ULONG ulGenCode, ULONG ulSubCode, char * szDescription, char *
szOperation ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

# hb\_errInternal()

## Syntax

C Prototype

```
#include <hbapierr.h>
hb_errInternal( ULONG ulIntCode, char * szText, char * szPar1, char * szPar2 ) -->
void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_errorHandler( )**

### **Syntax**

C Prototype

```
#include <hbapierr.h>
hb_errorHandler( HB_ERROR_INFO_PTR pNewHandler ) --> ( HB_ERROR_INFO_PTR )hParam
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

**hb\_fsChDir()**  
Change working directory

## Syntax

C Prototype

```
#include <hbapifs.h>
hb_fsChDir( BYTE * pszDirName ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_fsChDrv()**  
Change working drive

## Syntax

C Prototype

```
#include <hbapifs.h>
hb_fsChDrv( BYTE nDrive ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_fsClose()**

Close a file

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsClose( FHANDLE hFileHandle ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

**hb\_fsCommit()**  
Commit updates of a file

## Syntax

C Prototype

```
#include <hbapifs.h>
hb_fsCommit( FHANDLE hFileHandle ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_fsCreate()**  
Create a file

## Syntax

C Prototype

```
#include <hbapifs.h>
hb_fsCreate( BYTE * pszFileName, USHORT uiAttribute ) --> ( FHANDLE )hResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_fsCreateTemp( )**

Create a temporary file from components

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsCreateTemp( const BYTE * pszDir, const BYTE * pszPrefix, USHORT uiAttribute )
--> ( FHANDLE )hResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsCurDir()**

Retrieve a static pointer containing current directory for specified drive

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsCurDir( USHORT uiDrive ) --> ( BYTE * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsCurDirBuff()**

Copy current directory for given drive into a buffer

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsCurDirBuff( USHORT uiDrive, BYTE * pbyBuffer, ULONG ulLen ) --> ( USHORT
)usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsCurDrv( )**

Retrieve current drive number

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsCurDrv( void ) --> ( BYTE )cResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

**hb\_fsDelete()**  
Delete a file

## Syntax

C Prototype

```
#include <hbapifs.h>
hb_fsDelete( BYTE * pszFileName ) --> ( int )iResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_fsEof()**

Determine if an open file is position at end-of-file

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsEof( FHANDLE hFileHandle ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

**hb\_fsError()**  
Retrieve file system error

## Syntax

C Prototype

```
#include <hbapifs.h>
hb_fsError( void ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_fsFile()**

Determine if a file exists

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsFile( BYTE * pszFileName ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsFSize()**

Determine the size of a file

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsFSize( BYTE * pszFileName, BOOL bUseDirEntry ) --> ( ULONG )ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsExtOpen()**

Open a file using default extension and a list of paths

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsExtOpen( BYTE * pszFileName, BYTE * pDefExt, USHORT uiFlags, BYTE * pPaths,
PHB_ITEM pError ) --> ( FHANDLE )hResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsIsDrv()**

Determine if a drive number is a valid drive

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsIsDrv( BYTE nDrive ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsIsDevice()**

Determine if a file is attached to a device (console?)

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsIsDevice( FHANDLE hFileHandle ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsLock()**

Request a lock on a portion of a file

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsLock( FHANDLE hFileHandle, ULONG ulStart, ULONG ulLength, USHORT uiMode ) --> (
    BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

**hb\_fsMkDir()**  
Create a directory

## Syntax

C Prototype

```
#include <hbapifs.h>
hb_fsMkDir( BYTE * pszDirName ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_fsOpen()**  
Open a file

## Syntax

C Prototype

```
#include <hbapifs.h>
hb_fsOpen( BYTE * pszFileName, USHORT uiFlags ) --> ( FHANDLE )hResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_fsRead()**

Read contents of a file into a buffer (<=64K)

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsRead( FHANDLE hFileHandle, BYTE * pBuff, USHORT ulCount ) --> ( USHORT
)usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsReadLarge()**

Read contents of a file into a buffer (>64K)

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsReadLarge( FHANDLE hFileHandle, BYTE * pBuff, ULONG ulCount ) --> ( ULONG
)ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

**hb\_fsRmDir()**  
Remove a directory

## Syntax

C Prototype

```
#include <hbapifs.h>
hb_fsRmDir( BYTE * pszDirName ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_fsRename( )**  
Rename a file

## Syntax

C Prototype

```
#include <hbapifs.h>
hb_fsRename( BYTE * pszOldName, BYTE * pszNewName ) --> ( int )iResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_fsSeek()**

Reposition an open file

### **Syntax**

C Prototype

```
#include <hbapifs.h>
```

```
hb_fsSeek( FHANDLE hFileHandle, LONG lOffset, USHORT uiMode ) --> ( ULONG )ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsTell()**

Retrieve the current position of a file

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsTell( FHANDLE hFileHandle ) --> ( ULONG )ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsTempName( )**

Create a temporary file name in a buffer

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsTempName( BYTE * pszBuffer, const BYTE * pszDir, const BYTE * pszPrefix ) -->
void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsSetDevMode()**

Change the device mode of a file (text/binary)

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsSetDevMode( FHANDLE hFileHandle, USHORT uiDevMode ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsSetDevRaw()**

Change the device mode of a file to raw (binary)

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsSetDevRaw( FHANDLE hFileHandle ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsSetDevText()**

Change the device mode of a file to text

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsSetDevText( FHANDLE hFileHandle ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsSetError()**

Set the file system error number

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsSetError( USHORT uiError ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsWrite()**

Write to an open file from a buffer (<=64K)

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsWrite( FHANDLE hFileHandle, BYTE * pBuff, USHORT ulCount ) --> ( USHORT
)usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsWriteLarge()**

Write to an open file from a buffer (>64K)

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsWriteLarge( FHANDLE hFileHandle, BYTE * pBuff, ULONG ulCount ) --> ( ULONG
)ulResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsFNameSplit()**

Split given filename into path, name and extension

### **Syntax**

C Prototype

```
#include <hbapifs.h>
hb_fsFNameSplit( char * pszFileName ) --> ( PHB_FNAME )hResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_fsFNameMerge()**

This function joins path, name and extension into a string with a filename

### **Syntax**

C Prototype

```
#include <hbapifs.h>
```

```
hb_fsFNameMerge( char * pszFileName, PHB_FNAME pFileName ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

**hb\_fsFlock( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapifs.h>
```

```
hb_fsFlock( h, s, l ) --> <see hb_fsLock( h, s, l, FL_LOCK )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapifs.h

## Platforms

All

## **hb\_fsFUnlock()**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapifs.h>
hb_fsFUnlock( h, s, l ) --> <see hb_fsLock( h, s, l, FL_UNLOCK )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapifs.h

### **Platforms**

All

# hb\_gtInit()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtInit( int iFilenoStdin, int iFilenoStdout, int iFilenoStderr ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtExit()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtExit( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

# hb\_gtAdjustPos()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtAdjustPos( int iHandle, char * pStr, ULONG ulLen ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtBox( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtBox( USHORT uiTop, USHORT uiLeft, USHORT uiBottom, USHORT uiRight, BYTE *
pbyFrame ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## hb\_gtBoxD( )

### Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtBoxD( USHORT uiTop, USHORT uiLeft, USHORT uiBottom, USHORT uiRight ) --> (
USHORT )usResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is gt\* (ie. gtdos)

### Platforms

All

## hb\_gtBoxS( )

### Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtBoxS( USHORT uiTop, USHORT uiLeft, USHORT uiBottom, USHORT uiRight ) --> (
USHORT )usResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is gt\* (ie. gtdos)

### Platforms

All

**hb\_gtColorSelect()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtColorSelect( USHORT uiColorIndex ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtColorToN( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtColorToN( char * szColorString ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

# hb\_gtDispBegin( )

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtDispBegin( void ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtDispCount( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtDispCount( void ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

# hb\_gtDispEnd()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtDispEnd( void ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## **hb\_gtDrawShadow( )**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gtDrawShadow( USHORT uiTop, USHORT uiLeft, USHORT uiBottom, USHORT uiRight, BYTE
byAttr ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

**hb\_gtGetBlink()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtGetBlink( BOOL * pbBlink ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## hb\_gtGetColorStr()

### Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtGetColorStr( char * pszColorString ) --> ( USHORT )usResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is gt\* (ie. gtdos)

### Platforms

All

## **hb\_gtGetCursor( )**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gtGetCursor( USHORT * puiCursorShape ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

**hb\_gtGetPos( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtGetPos( SHORT * piRow, SHORT * piCol ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtIsColor()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtIsColor( void ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtMaxCol( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtMaxCol( void ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtMaxRow( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtMaxRow( void ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtPostExt()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtPostExt( void ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtPreExt( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtPreExt( void ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## **hb\_gtSuspend()**

Prepare the reminal for shell output

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gtSuspend( void ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_gtResume()**

Resume the terminal after the shell output

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gtResume( void ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## hb\_gtReadKey( )

### Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtReadKey( HB_inkey_enum eventmask ) --> ( int )iResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is gt\* (ie. gtdos)

### Platforms

All

## hb\_gtRectSize()

### Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtRectSize( USHORT uiTop, USHORT uiLeft, USHORT uiBottom, USHORT uiRight, USHORT
* puiBuffSize ) --> ( USHORT )usResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is gt\* (ie. gtdos)

### Platforms

All

## hb\_gtRepChar( )

### Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtRepChar( USHORT uiRow, USHORT uiCol, BYTE byChar, USHORT uiCount ) --> ( USHORT
)usResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is gt\* (ie. gtdos)

### Platforms

All

# hb\_gtRest()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtRest( USHORT uiTop, USHORT uiLeft, USHORT uiBottom, USHORT uiRight, void *
pScrBuff ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

# hb\_gtSave()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtSave( USHORT uiTop, USHORT uiLeft, USHORT uiBottom, USHORT uiRight, void *
pScrBuff ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtScrDim( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtScrDim( USHORT * puiHeight, USHORT * puiWidth ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

# hb\_gtScroll()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtScroll( USHORT uiTop, USHORT uiLeft, USHORT uiBottom, USHORT uiRight, SHORT
iRows, SHORT iCols ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtSetBlink( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtSetBlink( BOOL bBlink ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## hb\_gtSetColorStr()

### Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtSetColorStr( char * pszColorString ) --> ( USHORT )usResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is gt\* (ie. gtdos)

### Platforms

All

## **hb\_gtSetCursor()**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gtSetCursor( USHORT uiCursorShape ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

# hb\_gtSetMode()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtSetMode( USHORT uiRows, USHORT uiCols ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtSetPos( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtSetPos( SHORT iRow, SHORT iCol ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## **hb\_gtSetPosContext()**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gtSetPosContext( SHORT iRow, SHORT iCol, SHORT iMode ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_gtSetSnowFlag()**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gtSetSnowFlag( BOOL bNoSnow ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

**hb\_gtTone( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtTone( double dFrequency, double dDuration ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

# hb\_gtWrite()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtWrite( BYTE * pbyStr, ULONG ulLen ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## **hb\_gtWriteAt()**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gtWriteAt( USHORT uiRow, USHORT uiCol, BYTE * pbyStr, ULONG ulLen ) --> ( USHORT
)usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

# hb\_gtWriteCon()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtWriteCon( BYTE * pbyStr, ULONG ulLen ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

# hb\_gtVersion()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtVersion( void ) --> ( char * )pszResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

# hb\_gtWCreate()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtWCreate( HB_GT_RECT * rect, HB_GT_WND ** wnd ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

# hb\_gtWDestroy( )

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtWDestroy( HB_GT_WND * wnd ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtWFlash( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtWFlash( void ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtWApp( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtWApp( HB_GT_WND ** wnd ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtWCurrent( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtWCurrent( HB_GT_WND * wnd ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

# hb\_gtWPos( )

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtWPos( HB_GT_WND * wnd, HB_GT_RECT * rect ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

# hb\_gtWVis()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtWVis( HB_GT_WND * wnd, USHORT uiStatus ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtSLR()**  
System Level Request

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtSLR( HB_GT_SLR * pSLR ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtModalRead()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtModalRead( void * ) --> ( USHORT )usResult
```

## Arguments

<void \*>

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

# hb\_gtBeginWrite()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtBeginWrite( void ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtEndWrite()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtEndWrite( void ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## **hb\_gtFlushCursor()**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gtFlushCursor( void ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

# hb\_gtSetColor()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtSetColor( HB_GT_RGB * color ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gtGetColor()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gtGetColor( HB_GT_RGB * color ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## **hb\_gtSetBorder( )**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gtSetBorder( HB_GT_RGB * color ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## hb\_gt\_Init()

### Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_Init( int iFilenoStdin, int iFilenoStdout, int iFilenoStderr ) --> void
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is gt\* (ie. gtdos)

### Platforms

All

**hb\_gt\_Exit()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_Exit( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

# hb\_gt\_AdjustPos()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_AdjustPos( BYTE * pStr, ULONG ulLen ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gt\_Box( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_Box( USHORT uiTop, USHORT uiLeft, USHORT uiBottom, USHORT uiRight, BYTE *
pbyFrame, BYTE byAttr ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gt\_BoxD( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_BoxD( USHORT uiTop, USHORT uiLeft, USHORT uiBottom, USHORT uiRight, BYTE *
pbyFrame, BYTE byAttr ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## hb\_gt\_BoxS( )

### Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_BoxS( USHORT uiTop, USHORT uiLeft, USHORT uiBottom, USHORT uiRight, BYTE *
pbyFrame, BYTE byAttr ) --> ( USHORT )usResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is gt\* (ie. gtdos)

### Platforms

All

**hb\_gt\_Col()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_Col( void ) --> ( SHORT )sResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gt\_DispBegin( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_DispBegin( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gt\_DispCount( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_DispCount( void ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gt\_DispEnd( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_DispEnd( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gt\_GetBlink()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_GetBlink( void ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

# hb\_gt\_GetCursorStyle()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_GetCursorStyle( void ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gt\_GetScreenHeight()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_GetScreenHeight( void ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## **hb\_gt\_GetScreenWidth()**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gt_GetScreenWidth( void ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_gt\_GetText()**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gt_GetText( USHORT uiTop, USHORT uiLeft, USHORT uiBottom, USHORT uiRight, BYTE *
pbyDst ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_gt\_HorizLine()**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gt_HorizLine( USHORT uiRow, USHORT uiLeft, USHORT uiRight, BYTE byChar, BYTE
byAttr ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

**hb\_gt\_IsColor()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_IsColor( void ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gt\_PreExt()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_PreExt( void ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gt\_PostExt( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_PostExt( void ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## **hb\_gt\_Suspend()**

Suspend the terminal before the shell call

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gt_Suspend( void ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_gt\_Resume()**

Resume the terminal after the shell call

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gt_Resume( void ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_gt\_Puts()**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gt_Puts( USHORT uiRow, USHORT uiCol, BYTE byAttr, BYTE * pbyStr, ULONG ullLen )
--> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_gt\_PutText( )**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gt_PutText( USHORT uiTop, USHORT uiLeft, USHORT uiBottom, USHORT uiRight, BYTE *
pbySrc ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_gt\_ReadKey( )**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gt_ReadKey( HB_inkey_enum eventmask ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_gt\_RectSize()**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gt_RectSize( USHORT rows, USHORT cols ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

# hb\_gt\_Replicate()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_Replicate( USHORT uiTop, USHORT uiLeft, BYTE byAttr, BYTE byChar, ULONG ullLen
) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gt\_Row( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_Row( void ) --> ( SHORT )sResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## hb\_gt\_Scroll()

### Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_scroll( USHORT uiTop, USHORT uiLeft, USHORT uiBottom, USHORT uiRight, BYTE
byAttr, SHORT iRows, SHORT iCols ) --> void
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is gt\* (ie. gtdos)

### Platforms

All

# hb\_gt\_SetAttribute()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_SetAttribute( USHORT uiTop, USHORT uiLeft, USHORT uiBottom, USHORT uiRight,
BYTE byAttr ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gt\_SetBlink()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_SetBlink( BOOL bBlink ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

# hb\_gt\_SetCursorStyle()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_SetCursorStyle( USHORT uiCursorShape ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gt\_SetMode( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_SetMode( USHORT uiRows, USHORT uiCols ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## **hb\_gt\_SetPos()**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gt_SetPos( SHORT iRow, SHORT iCol, SHORT iMethod ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

**hb\_gt\_Tone( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_Tone( double dFrequency, double dDuration ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_gt\_Version()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_gt_Version( void ) --> ( char * )pszResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## **hb\_gt\_VertLine()**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_gt_VertLine( USHORT uiCol, USHORT uiTop, USHORT uiBottom, BYTE byChar, BYTE
byAttr ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_inkey()**

Wait for keyboard input

### **Syntax**

C Prototype

```
#include <hbapigt.h>
```

```
hb_inkey( BOOL bWait, double dSeconds, HB_inkey_enum event_mask ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_inkeyGet()**

Extract the next key from the Harbour keyboard buffer

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_inkeyGet( void ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_inkeyPut()**

Inserts an inkey code into the keyboard buffer

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_inkeyPut( int ch ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_inkeyLast()**

Return the value of the last key that was extracted

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_inkeyLast( void ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_inkeyNext()**

Return the next key without extracting it

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_inkeyNext( void ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_inkeyPoll()**

Poll the console keyboard to stuff the Harbour buffer

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_inkeyPoll( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_inkeyReset()**

Reset the Harbour keyboard buffer

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_inkeyReset( BOOL allocate ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

**hb\_mouseIsPresent( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouseIsPresent( void ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_mouseGetCursor( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouseGetCursor( void ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_mouseSetCursor( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouseSetCursor( BOOL bVisible ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_mouseCol()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouseCol( void ) --> ( int )iResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_mouseRow( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouseRow( void ) --> ( int )iResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_mouseSetPos( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouseSetPos( int iRow, int iCol ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## **hb\_mouseIsButtonPressed( )**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_mouseIsButtonPressed( int iButton ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

**hb\_mouseCountButton( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouseCountButton( void ) --> ( int )iResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_mouseSetBounds( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouseSetBounds( int iTop, int iLeft, int iBottom, int iRight ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## **hb\_mouseGetBounds( )**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_mouseGetBounds( int * piTop, int * piLeft, int * piBottom, int * piRight ) -->
void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

**hb\_mouse\_Init()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouse_Init( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_mouse\_Exit()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouse_Exit( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_mouse\_IsPresent()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouse_IsPresent( void ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_mouse\_Show( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouse_Show( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_mouse\_Hide()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouse_Hide( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_mouse\_Col()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouse_Col( void ) --> ( int )iResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_mouse\_Row( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouse_Row( void ) --> ( int )iResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_mouse\_SetPos( )**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouse_SetPos( int iRow, int iCol ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

**hb\_mouse\_IsButtonPressed()**

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_mouse_IsButtonPressed( int iButton ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## **hb\_mouse\_CountButton()**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_mouse_CountButton( void ) --> ( int )iResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_mouse\_SetBounds( )**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_mouse_SetBounds( int iTop, int iLeft, int iBottom, int iRight ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

## **hb\_mouse\_GetBounds( )**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_mouse_GetBounds( int * piTop, int * piLeft, int * piBottom, int * piRight ) -->
void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

# hb\_setkeyInit()

## Syntax

C Prototype

```
#include <hbapigt.h>
hb_setkeyInit( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is gt\* (ie. gtdos)

## Platforms

All

## **hb\_setkeyExit()**

### **Syntax**

C Prototype

```
#include <hbapigt.h>
hb_setkeyExit( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is gt\* (ie. gtdos)

### **Platforms**

All

**hb\_evalLaunch( )**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_evalLaunch( PEVALINFO pEvalInfo ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## hb\_evalNew( )

### Syntax

C Prototype

```
#include <hbapiitm.h>
hb_evalNew( PEVALINFO pEvalInfo, PHB_ITEM pItem ) --> ( BOOL )bResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is rtl

### Platforms

All

**hb\_evalPutParam( )**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_evalPutParam( PEVALINFO pEvalInfo, PHB_ITEM pItem ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_evalRelease()**

### **Syntax**

C Prototype

```
#include <hbapiitm.h>
hb_evalRelease( PEVALINFO pEvalInfo ) --> ( BOOL )bResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## hb\_itemDo()

### Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemDo( PHB_ITEM pItem, USHORT uiPCount, PHB_ITEM pItemArg1, ... ) --> ( PHB_ITEM
)pResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is rtl

### Platforms

All

## hb\_itemDoC()

### Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemDoC( char * szFunc, USHORT uiPCount, PHB_ITEM pItemArg1, ... ) --> ( PHB_ITEM
)pResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is rtl

### Platforms

All

**hb\_itemArrayGet()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemArrayGet( PHB_ITEM pArray, ULONG ulIndex ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemArrayNew( )**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemArrayNew( ULONG ulLen ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## hb\_itemArrayPut()

### Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemArrayPut( PHB_ITEM pArray, ULONG ulIndex, PHB_ITEM pItem ) --> ( PHB_ITEM
)pResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is rtl

### Platforms

All

# hb\_itemCopyC()

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemCopyC( PHB_ITEM pItem, char * szBuffer, ULONG ulLen ) --> ( ULONG )ulResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## hb\_itemFreeC()

### Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemFreeC( char * szText ) --> ( BOOL )bResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is rtl

### Platforms

All

**hb\_itemGetC()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemGetC( PHB_ITEM pItem ) --> ( char * )pszResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_itemGetCPtr()**

### **Syntax**

C Prototype

```
#include <hbapiitm.h>
hb_itemGetCPtr( PHB_ITEM pItem ) --> ( char * )pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

**hb\_itemGetCLen()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemGetCLen( PHB_ITEM pItem ) --> ( ULONG )ulResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemGetDS()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemGetDS( PHB_ITEM pItem, char * szDate ) --> ( char * )pszResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemGetDL()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemGetDL( PHB_ITEM pItem ) --> ( long )lResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemGetL()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemGetL( PHB_ITEM pItem ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemGetND()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemGetND( PHB_ITEM pItem ) --> ( double )dResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemGetNI()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemGetNI( PHB_ITEM pItem ) --> ( int )iResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemGetNL()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemGetNL( PHB_ITEM pItem ) --> ( long )lResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## hb\_itemGetNLen()

### Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemGetNLen( PHB_ITEM pItem, int * piWidth, int * piDec ) --> void
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is rtl

### Platforms

All

**hb\_itemGetPtr()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemGetPtr( PHB_ITEM pItem ) --> ( void * )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## hb\_itemNew()

### Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemNew( PHB_ITEM pNull ) --> ( PHB_ITEM )pResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is rtl

### Platforms

All

**hb\_itemInit()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemInit( PHB_ITEM pItem ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemPCount()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemPCount( void ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_itemParam()**

Creates a copy of an item parameter (outside the eval stack)

### **Syntax**

C Prototype

```
#include <hbapiitm.h>
hb_itemParam( USHORT uiParam ) --> ( PHB_ITEM )pResult
```

### **Arguments**

**<uiParam>**    The 1-based parameter to copy and retrieve.

### **Returns**

### **Description**

Use this function whenever the pointer needs to be accessed after the current function returns; for example, if the pointer is to be copied to a static variable or structure member for later access.

Compare to `hb_param()`, which simply gets a direct pointer to the item on the stack.

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

### **See Also:**

[hb\\_param\(\)](#)

**hb\_itemPutC()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemPutC( PHB_ITEM pItem, char * szText ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## hb\_itemPutCPtr()

### Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemPutCPtr( PHB_ITEM pItem, char * szText, ULONG ulLen ) --> ( PHB_ITEM )pResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is rtl

### Platforms

All

## hb\_itemPutCL()

### Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemPutCL( PHB_ITEM pItem, char * szText, ULONG ulLen ) --> ( PHB_ITEM )pResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is rtl

### Platforms

All

**hb\_itemSetCMemo( )**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemSetCMemo( PHB_ITEM pItem ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemPutD()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemPutD( PHB_ITEM pItem, long lYear, long lMonth, long lDay ) --> ( PHB_ITEM
)pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemPutDS()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemPutDS( PHB_ITEM pItem, char * szDate ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemPutDL()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemPutDL( PHB_ITEM pItem, long lJulian ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemPutL()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemPutL( PHB_ITEM pItem, BOOL bValue ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemPutND()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemPutND( PHB_ITEM pItem, double dNumber ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemPutNI()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemPutNI( PHB_ITEM pItem, int iNumber ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemPutNL()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemPutNL( PHB_ITEM pItem, long lNumber ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_itemPutNLen()**

### **Syntax**

C Prototype

```
#include <hbapiitm.h>
hb_itemPutNLen( PHB_ITEM pItem, double dNumber, int iWidth, int iDec ) --> (
PHB_ITEM )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

**hb\_itemPutNDLen( )**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemPutNDLen( PHB_ITEM pItem, double dNumber, int iWidth, int iDec ) --> (
PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemPutNLen( )**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemPutNLen( PHB_ITEM pItem, int iNumber, int iWidth ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemPutNLLen( )**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemPutNLLen( PHB_ITEM pItem, long lNumber, int iWidth ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemPutPtr()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemPutPtr( PHB_ITEM pItem, void * pValue ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

# hb\_itemRelease()

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemRelease( PHB_ITEM pItem ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemReturn()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemReturn( PHB_ITEM pItem ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemSize()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemSize( PHB_ITEM pItem ) --> ( ULONG )ulResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemType()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemType( PHB_ITEM pItem ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

# hb\_itemTypeStr()

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemTypeStr( PHB_ITEM pItem ) --> ( char * )pszResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemParamPtr()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemParamPtr( USHORT uiParam, int iMask ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemReturnPtr()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemReturnPtr( void ) --> ( PHB_ITEM )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_itemStrCmp()**  
Our string compare

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemStrCmp( PHB_ITEM pFirst, PHB_ITEM pSecond, BOOL bForceExact ) --> ( int )iResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_itemCopy()**

Copies an item to one place to another respecting its contents

### **Syntax**

C Prototype

```
#include <hbapiitm.h>
```

```
hb_itemCopy( PHB_ITEM pDest, PHB_ITEM pSource ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

**hb\_itemClear()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemClear( PHB_ITEM pItem ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## **hb\_itemUnRef()**

De-references passed variable

### **Syntax**

C Prototype

```
#include <hbapiitm.h>
```

```
hb_itemUnRef( PHB_ITEM pItem ) --> ( PHB_ITEM )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_itemStr()**

Convert a number to a string

### **Syntax**

C Prototype

```
#include <hbapiitm.h>
hb_itemStr( PHB_ITEM pNumber, PHB_ITEM pWidth, PHB_ITEM pDec ) --> ( char *
)pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_itemString()**

Convert any scalar to a string

### **Syntax**

C Prototype

```
#include <hbapiitm.h>
hb_itemString( PHB_ITEM pItem, ULONG * uLen, BOOL * bFreeReq ) --> ( char *
)pszResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## **hb\_itemValToStr()**

Convert any scalar to a string

### **Syntax**

C Prototype

```
#include <hbapiitm.h>
```

```
hb_itemValToStr( PHB_ITEM pItem ) --> ( PHB_ITEM )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

## hb\_itemPadConv()

### Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemPadConv( PHB_ITEM pItem, char * buffer, ULONG * pulSize ) --> ( char *
)pszResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is rtl

### Platforms

All

**hb\_itemSwap()**

## Syntax

C Prototype

```
#include <hbapiitm.h>
hb_itemSwap( PHB_ITEM pItem1, PHB_ITEM pItem2 ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

# hb\_langRegister()

## Syntax

C Prototype

```
#include <hbapilng.h>
hb_langRegister( PHB_LANG lang ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is lang

## Platforms

All

# hb\_langDeRegister()

## Syntax

C Prototype

```
#include <hbapilng.h>
hb_langDeRegister( char * pszID ) --> ( BOOL )bResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is lang

## Platforms

All

**hb\_langFind( )**

## Syntax

C Prototype

```
#include <hbapilng.h>
hb_langFind( char * pszID ) --> ( PHB_LANG )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is lang

## Platforms

All

**hb\_langSelect()**

## Syntax

C Prototype

```
#include <hbapilng.h>
hb_langSelect( PHB_LANG lang ) --> ( PHB_LANG )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is lang

## Platforms

All

**hb\_langSelectID()**

## Syntax

C Prototype

```
#include <hbapilng.h>
hb_langSelectID( char * pszID ) --> ( char * )pszResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is lang

## Platforms

All

**hb\_langDGetItem( )**

## Syntax

C Prototype

```
#include <hbapilng.h>
hb_langDGetItem( int iIndex ) --> ( void * )pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is lang

## Platforms

All

**hb\_langID( )**

## Syntax

C Prototype

```
#include <hbapilng.h>
hb_langID( void ) --> ( char * )pszResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is lang

## Platforms

All

**hb\_langName( )**

## Syntax

C Prototype

```
#include <hbapilng.h>
hb_langName( void ) --> ( char * )pszResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is lang

## Platforms

All

## hb\_langDGetErrorDesc()

### Syntax

C Prototype

```
#include <hbapilng.h>
hb_langDGetErrorDesc( ULONG ulIndex ) --> ( char * )pszResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is lang

### Platforms

All

## HB\_LANG\_REQUEST( )

### Syntax

C Prototype (macro definition)

```
#include <hbapilng.h>
HB_LANG_REQUEST( id ) --> <see HB_LANG_REQUEST_( id )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapilng.h

### Platforms

All

## **hb\_rddInsertAreaNode( )**

### **Syntax**

C Prototype

```
#include <hbapirdd.h>
hb_rddInsertAreaNode( char *szDriver ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rdd, nulsys, dbfntx, dbfcdx

### **Platforms**

All

**hb\_rddGetCurrentWorkAreaNumber( )**

## Syntax

C Prototype

```
#include <hbapirdd.h>
hb_rddGetCurrentWorkAreaNumber( void ) --> ( int )iResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rdd, nulsys, dbfntx, dbfcdx

## Platforms

All

## **hb\_rddGetCurrentWorkAreaPointer()**

### **Syntax**

C Prototype

```
#include <hbapirdd.h>
hb_rddGetCurrentWorkAreaPointer( void ) --> ( void * )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rdd, nulsys, dbfntx, dbfcdx

### **Platforms**

All

## **hb\_rddSelectWorkAreaAlias()**

### **Syntax**

C Prototype

```
#include <hbapirdd.h>
hb_rddSelectWorkAreaAlias( char * szAlias ) --> ( ERRCODE )hResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rdd, nulsys, dbfntx, dbfcdx

### **Platforms**

All

## **hb\_rddSelectWorkAreaNumber( )**

### **Syntax**

C Prototype

```
#include <hbapirdd.h>
hb_rddSelectWorkAreaNumber( int iArea ) --> ( ERRCODE )hResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rdd, nulsys, dbfntx, dbfcdx

### **Platforms**

All

**hb\_rddSelectWorkAreaSymbol()**

## Syntax

C Prototype

```
#include <hbapirdd.h>
hb_rddSelectWorkAreaSymbol( PHB_SYMB pSymAlias ) --> ( ERRCODE )hResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rdd, nulsys, dbfntx, dbfcdx

## Platforms

All

## hb\_rddGetFieldValue()

### Syntax

C Prototype

```
#include <hbapirdd.h>
hb_rddGetFieldValue( HB_ITEM_PTR pItem, PHB_SYMB pFieldSymbol ) --> ( ERRCODE
)hResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is rdd, nulsys, dbfntx, dbfcdx

### Platforms

All

# hb\_rddPutFieldValue()

## Syntax

C Prototype

```
#include <hbapirdd.h>
hb_rddPutFieldValue( HB_ITEM_PTR pItem, PHB_SYMB pFieldSymbol ) --> ( ERRCODE
)hResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rdd, nulsys, dbfntx, dbfcdx

## Platforms

All

## hb\_rddFieldGet()

### Syntax

C Prototype

```
#include <hbapirdd.h>
hb_rddFieldGet( HB_ITEM_PTR pItem, PHB_SYMB pFieldSymbol ) --> ( ERRCODE )hResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is rdd, nulsys, dbfntx, dbfcdx

### Platforms

All

# hb\_rddFieldPut()

## Syntax

C Prototype

```
#include <hbapirdd.h>
hb_rddFieldPut( HB_ITEM_PTR pItem, PHB_SYMB pFieldSymbol ) --> ( ERRCODE )hResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rdd, nulsys, dbfntx, dbfcdx

## Platforms

All

**hb\_rddShutDown( )**

## Syntax

C Prototype

```
#include <hbapirdd.h>
hb_rddShutDown( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rdd, nulsys, dbfntx, dbfcdx

## Platforms

All

## hb\_rddInherit()

### Syntax

C Prototype

```
#include <hbapirdd.h>
hb_rddInherit( PRDDFUNCS pTable, PRDDFUNCS pSubTable, PRDDFUNCS pSuperTable, BYTE *
szDrvName ) --> ( ERRCODE )hResult
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Library is rdd, nulsys, dbfntx, dbfcdx

### Platforms

All

**hb\_rddDisinherit()**

## Syntax

C Prototype

```
#include <hbapirdd.h>
hb_rddDisinherit( BYTE * drvName ) --> ( ERRCODE )hResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rdd, nulsys, dbfntx, dbfcdx

## Platforms

All

## **hb\_rddExtendType( )**

### **Syntax**

C Prototype

```
#include <hbapirdd.h>
hb_rddExtendType( USHORT fieldType ) --> ( USHORT )usResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is rdd, nulsys, dbfntx, dbfcdx

### **Platforms**

All

**hb\_rddFieldType()**

## Syntax

C Prototype

```
#include <hbapirdd.h>
hb_rddFieldType( USHORT extendType ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rdd, nulsys, dbfntx, dbfcdx

## Platforms

All

## SELF\_BOF( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_BOF( w, sp ) --> <see ( ( *( w )->lprfsHost->bof )( w, sp ) )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

**SELF\_EOF( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_EOF( w, sp ) --> <see ( ( *( w )->lprfsHost->eof )( w, sp ) )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapirdd.h

## Platforms

All

## **SELF\_FOUND( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_FOUND( w, sp ) --> <see ( ( *( w )->lprfsHost->found )( w, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

**SELF\_GOTO( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_GOTO( w, l ) --> <see ( ( *( w )->lprfsHost->go )( w, l ) )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapirdd.h

## Platforms

All

**SELF\_GOTOID( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_GOTOID( w, sp ) --> <see ( ( *( w )->lprfsHost->goToId )( w, sp ) )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapirdd.h

## Platforms

All

**SELF\_GOBOTTOM( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_GOBOTTOM( w ) --> <see ( ( *( w )->lprfsHost->goBottom )( w ) )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapirdd.h

## Platforms

All

## **SELF\_GOTOP( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_GOTOP( w ) --> <see ( ( *( w )->lprfsHost->goTop )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_SEEK( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_SEEK( w, i1, v, i2 ) --> <see ( ( *( w )->lprfsHost->seek )( w, i1, v, i2 ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

**SELF\_SKIP()**

## Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_SKIP( w, l ) --> <see ( ( *( w )->lprfsHost->skip )( w, l ) )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapirdd.h

## Platforms

All

## **SELF\_SKIPFILTER( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_SKIPFILTER( w, l ) --> <see ( ( *( w )->lprfsHost->skipFilter )( w, l ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SELF\_SKIPRAW( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_SKIPRAW( w, l ) --> <see ( ( *( w )->lprfsHost->skipRaw )( w, l ) )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## **SELF\_ADDFIELD( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_ADDFIELD( w, ip ) --> <see ( ( *( w )->lprfsHost->addField )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

**SELF\_APPEND( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_APPEND( w, l ) --> <see ( ( *( w )->lprfsHost->append )( w, l ) )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapirdd.h

## Platforms

All

## **SELF\_CREATEFIELDS( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_CREATEFIELDS( w, v ) --> <see ( ( *( w )->lprfsHost->createFields )( w, v ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_DELETE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_DELETE( w ) --> <see ( ( *( w )->lprfsHost->deleterec )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_DELETED( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_DELETED( w, sp ) --> <see ( ( *( w )->lprfsHost->deleted )( w, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_FIELDCOUNT( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_FIELDCOUNT( w, sp ) --> <see ( ( *( w )->lprfsHost->fieldCount )( w, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_FIELDDISPLAY( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_FIELDDISPLAY( w, sp ) --> <see ( ( *( w )->lprfsHost->fieldDisplay )( w, sp )
)>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_FIELDINFO()**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_FIELDINFO( w, s1, s2, v ) --> <see ( ( *( w )->lprfsHost->fieldInfo )( w, s1,
s2, v ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_FIELDNAME( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_FIELDNAME( w, i, bp ) --> <see ( ( *( w )->lprfsHost->fieldName )( w, i, bp )
)>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_FLUSH( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_FLUSH( w ) --> <see ( ( *( w )->lprfsHost->flush )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_GETREC( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_GETREC( w, bpp ) --> <see ( ( *( w )->lprfsHost->getRec )( w, bpp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_GETVALUE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_GETVALUE( w, i, v ) --> <see ( ( *( w )->lprfsHost->getValue )( w, i, v ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_GETVARLEN( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_GETVARLEN( w, i, lp ) --> <see ( ( *( w )->lprfsHost->getVarLen )( w, i, lp )
)>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

**SELF\_GOCOLD( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_GOCOLD( w ) --> <see ( ( *( w )->lprfsHost->goCold )( w ) )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapirdd.h

## Platforms

All

## **SELF\_GOHOT( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_GOHOT( w ) --> <see ( ( *( w )->lprfsHost->goHot )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SELF\_PUTVALUE( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_PUTVALUE( w, i, v ) --> <see ( ( *( w )->lprfsHost->putValue )( w, i, v ) )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## **SELF\_PUTREC( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_PUTREC( w, bp ) --> <see ( ( *( w )->lprfsHost->putRec )( w, bp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_RECALL( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_RECALL( w ) --> <see ( ( *( w )->lprfsHost->recall )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SELF\_RECCOUNT( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_RECCOUNT( w, sp ) --> <see ( ( *( w )->lprfsHost->reccount )( w, sp ) )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_RECINFO()

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_RECINFO( w, v1, i, v2 ) --> <see ( ( *( w )->lprfsHost->recInfo )( w, v1, i, v2 ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## **SELF\_RECNO( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_RECNO( w, i ) --> <see ( ( *( w )->lprfsHost->recno )( w, i ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_SETFIELDEXTENT( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_SETFIELDEXTENT( w, s ) --> <see ( ( *( w )->lprfsHost->setFieldExtent )( w, s )
)>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_ALIAS( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_ALIAS( w, bp ) --> <see ( ( *( w )->lprfsHost->alias )( w, bp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_CLOSE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_CLOSE( w ) --> <see ( ( *( w )->lprfsHost->close )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_CREATE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_CREATE( w, ip ) --> <see ( ( *( w )->lprfsHost->create )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_INFO()**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_INFO( w, i, g ) --> <see ( ( *( w )->lprfsHost->info )( w, i, g ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_NEW( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_NEW( w ) --> <see ( ( *( w )->lprfsHost->newarea )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

**SELF\_OPEN( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_OPEN( w, ip ) --> <see ( ( *( w )->lprfsHost->open )( w, ip ) )>
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Header file is hbapirdd.h

## Platforms

All

## **SELF\_RELEASE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_RELEASE( w ) --> <see ( ( *( w )->lprfsHost->release )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_STRUCTSIZE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_STRUCTSIZE( w, sp ) --> <see ( ( *( w )->lprfsHost->structSize )( w, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_SYSNAME( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_SYSNAME( w, bp ) --> <see ( ( *( w )->lprfsHost->sysName )( w, bp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

**SELF\_DBEVAL( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_DBEVAL( w, ip ) --> <see ( ( *( w )->lprfsHost->dbEval )( w, ip ) )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapirdd.h

## Platforms

All

## **SELF\_PACK( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_PACK( w ) --> <see ( ( *( w )->lprfsHost->pack )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SELF\_PACKREC( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_PACKREC( w, l, sp ) --> <see ( ( *( w )->lprfsHost->packRec )( w, l, sp ) )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## **SELF\_SORT( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_SORT( w, ip ) --> <see ( ( *( w )->lprfsHost->sort )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_TRANS( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_TRANS( w, ip ) --> <see ( ( *( w )->lprfsHost->trans )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_TRANSREC( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_TRANSREC( w, ip ) --> <see ( ( *( w )->lprfsHost->transRec )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_ZAP( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_ZAP( w ) --> <see ( ( *( w )->lprfsHost->zap )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_CHILDEND( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_CHILDEND( w, ip ) --> <see ( ( *( w )->lprfsHost->childEnd )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_CHILDSTART( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_CHILDSTART( w, ip ) --> <see ( ( *( w )->lprfsHost->childStart )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_CHILDSYNC( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_CHILDSYNC( w, ip ) --> <see ( ( *( w )->lprfsHost->childSync )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_SYNCCHILDREN( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_SYNCCHILDREN( w ) --> <see ( ( *( w )->lprfsHost->syncChildren )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_CLEARREL( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_CLEARREL( w ) --> <see ( ( *( w )->lprfsHost->clearRel )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_FORCEREL( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_FORCEREL( w ) --> <see ( ( *( w )->lprfsHost->forceRel )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_RELAREA( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_RELAREA( w, s, sp ) --> <see ( ( *( w )->lprfsHost->relArea )( w, s, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_RELEVAL( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_RELEVAL( w, ip ) --> <see ( ( *( w )->lprfsHost->relEval )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_RELTEXT( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_RELTEXT( w, s, bp ) --> <see ( ( *( w )->lprfsHost->relText )( w, s, bp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_SETREL( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_SETREL( w, ip ) --> <see ( ( *( w )->lprfsHost->setRel )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SELF\_ORDLSTADD( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_ORDLSTADD( w, lp ) --> <see ( ( *( w )->lprfsHost->orderListAdd )( w, lp ) )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_ORDLSTDELETE( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_ORDLSTDELETE( w, lp ) --> <see ( ( *( w )->lprfsHost->orderListDelete )( w, lp
) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_ORDLSTFOCUS( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_ORDLSTFOCUS( w, lp ) --> <see ( ( *( w )->lprfsHost->orderListFocus )( w, lp )
)>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## **SELF\_ORDLSTREBUILD( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_ORDLSTREBUILD( w ) --> <see ( ( *( w )->lprfsHost->orderListRebuild )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SELF\_ORDLSTCLEAR( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_ORDLSTCLEAR( w ) --> <see ( ( *( w )->lprfsHost->orderListClear )( w ) )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_ORDSETCOND( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_ORDSETCOND( w, ip ) --> <see ( ( *( w )->lprfsHost->orderCondition )( w, ip )
)>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## **SELF\_ORDCREATE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_ORDCREATE( w, ip ) --> <see ( ( *( w )->lprfsHost->orderCreate )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SELF\_ORDDESTROY( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_ORDDESTROY( w, p ) --> <see ( ( *( w )->lprfsHost->orderDestroy )( w, p ) )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_ORDINFO()

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_ORDINFO( w, i, p ) --> <see ( ( *( w )->lprfsHost->orderInfo )( w, i, p ) )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_ORDEXPR( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_ORDEXPR( w, p ) --> <see ( ( *( w )->lprfsHost->orderInfo )( w,
DBOI_EXPRESSION, p ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_ORDCOND( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_ORDCOND( w, p ) --> <see ( ( *( w )->lprfsHost->orderInfo )( w, DBOI_CONDITION,
p ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_ORDRECNO( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_ORDRECNO( w, p ) --> <see ( ( *( w )->lprfsHost->orderInfo )( w, DBOI_RECNO, p
) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

**SELF\_ORDPOS ( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_ORDPOS( w, p ) --> <see ( ( *( w )->lprfsHost->orderInfo )( w, DBOI_POSITION,
p ) )>
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Header file is hbapirdd.h

## Platforms

All

## SELF\_ORDNUMBER( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_ORDNUMBER( w, p ) --> <see ( ( *( w )->lprfsHost->orderInfo )( w, DBOI_NUMBER,
p ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_ORDNAME( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_ORDNAME( w, p ) --> <see ( ( *( w )->lprfsHost->orderInfo )( w, DBOI_NAME, p ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_ORDBAGNAME( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_ORDBAGNAME( w, p ) --> <see ( ( *( w )->lprfsHost->orderInfo )( w,
DBOI_BAGNAME,p ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_ORDBAGEXT( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_ORDBAGEXT( w, p ) --> <see ( ( *( w )->lprfsHost->orderInfo )( w, DBOI_BAGEXT,
p ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## **SELF\_CLEARFILTER( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_CLEARFILTER( w ) --> <see ( ( *( w )->lprfsHost->clearFilter )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_CLEARLOCATE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_CLEARLOCATE( w ) --> <see ( ( *( w )->lprfsHost->clearLocate )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_CLEARSCOPE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_CLEARSCOPE( w ) --> <see ( ( *( w )->lprfsHost->clearScope )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_COUNTSCOPE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_COUNTSCOPE( w, ip, lp ) --> <see ( ( *( w )->lprfsHost->countScope )( w, ip, lp
) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_FILTERTEXT( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_FILTERTEXT( w, bp ) --> <see ( ( *( w )->lprfsHost->filterText )( w, bp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SELF\_SCOPEINFO()

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_SCOPEINFO( w, i, v ) --> <see ( ( *( w )->lprfsHost->scopeInfo )( w, i, v ) )>
```

### Arguments

### Returns

### Description

### Status

Ready

Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## **SELF\_SETFILTER( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_SETFILTER( w, ip ) --> <see ( ( *( w )->lprfsHost->setFilter )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_SETLOCATE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_SETLOCATE( w, ip ) --> <see ( ( *( w )->lprfsHost->setLocate )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_SETSCOPE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_SETSCOPE( w, ip ) --> <see ( ( *( w )->lprfsHost->setScope )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SELF\_SKIPSCOPE( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_SKIPSCOPE( w, bp, l ) --> <see ( ( *( w )->lprfsHost->skipScope )( w, bp, l )
)>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_COMPILE()

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_COMPILE( w, bp ) --> <see ( ( *( w )->lprfsHost->compile )( w, bp ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## **SELF\_ERROR( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_ERROR( w, ip ) --> <see ( ( *( w )->lprfsHost->error )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_EVALBLOCK( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_EVALBLOCK( w, v ) --> <see ( ( *( w )->lprfsHost->evalBlock )( w, v ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SELF\_GETLOCKS( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_GETLOCKS( w, g ) --> <see ( ( *( w )->lprfsHost->info )( w, DBI_GETLOCKARRAY, g
) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## **SELF\_RAWLOCK( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_RAWLOCK( w, i, l ) --> <see ( ( *( w )->lprfsHost->rawlock )( w, i, l ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_LOCK( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_LOCK( w, sp ) --> <see ( ( *( w )->lprfsHost->lock )( w, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_UNLOCK( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_UNLOCK( w, l ) --> <see ( ( *( w )->lprfsHost->unlock )( w, l ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_CLOSEMEMFILE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_CLOSEMEMFILE( w ) --> <see ( ( *( w )->lprfsHost->closeMemFile )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_CREATEMEMFILE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_CREATEMEMFILE( w, bp ) --> <see ( ( *( w )->lprfsHost->createMemFile )( w, bp )
)>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SELF\_GETVALUEFILE( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_GETVALUEFILE( w, i, bp ) --> <see ( ( *( w )->lprfsHost->getValueFile )( w, i,
bp ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## **SELF\_OPENMEMFILE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SELF_OPENMEMFILE( w, bp ) --> <see ( ( *( w )->lprfsHost->openMemFile )( w, bp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SELF\_PUTVALUEFILE( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_PUTVALUEFILE( w, i, bp ) --> <see ( ( *( w )->lprfsHost->putValueFile )( w, i,
bp ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## **SELF\_READDBHEADER( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_READDBHEADER( w ) --> <see ( ( *( w )->lprfsHost->readDBHeader )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SELF\_WRITEDBHEADER( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_WRITEDBHEADER( w ) --> <see ( ( *( w )->lprfsHost->writeDBHeader )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SELF\_RECSIZE( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_RECSIZE( w, lp ) --> <see ( ( *( w )->lprfsHost->info )( w, DBI_GETRECSIZE, lp
) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_HEADERSIZE( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_HEADERSIZE( w, fp ) --> <see ( ( *( w )->lprfsHost->info )( w,
DBI_GETHEADERSIZE, fp ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_LUPDATE( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_LUPDATE( w, fp ) --> <see ( ( *( w )->lprfsHost->info )( w, DBI_LASTUPDATE, fp
) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_SETDELIM( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_SETDELIM( w, fp ) --> <see ( ( *( w )->lprfsHost->info )( w, DBI_SETDELIMITER,
fp ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_GETDELIM( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_GETDELIM( w, fp ) --> <see ( ( *( w )->lprfsHost->info )( w, DBI_GETDELIMITER,
fp ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SELF\_TABLEEXT( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SELF_TABLEEXT( w, fp ) --> <see ( ( *( w )->lprfsHost->info )( w, DBI_TABLEEXT, fp )
)>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## **SUPER\_BOF( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_BOF( w, sp ) --> <see ( ( *( SUPERTABLE )->bof )( w, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_EOF( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_EOF( w, sp ) --> <see ( ( *( SUPERTABLE )->eof )( w, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_FOUND( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_FOUND( w, sp ) --> <see ( ( *( SUPERTABLE )->found )( w, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_GOTO( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_GOTO( w, l ) --> <see ( ( *( SUPERTABLE )->go )( w, l ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_GOTOID( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_GOTOID( w, sp ) --> <see ( ( *( SUPERTABLE )->goToId )( w, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_GOBOTTOM( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_GOBOTTOM( w ) --> <see ( ( *( SUPERTABLE )->goBottom )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_GOTOP( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_GOTOP( w ) --> <see ( ( *( SUPERTABLE )->goTop )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_SEEK( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_SEEK( w, i1, v, i2 ) --> <see ( ( *( SUPERTABLE )->seek )( w, i1, v, i2 ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_SKIP( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_SKIP( w, l ) --> <see ( ( *( SUPERTABLE )->skip )( w, l ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_SKIPFILTER( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_SKIPFILTER( w, l ) --> <see ( ( *( SUPERTABLE )->skipFilter )( w, l ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_SKIPRAW( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_SKIPRAW( w, l ) --> <see ( ( *( SUPERTABLE )->skipRaw )( w, l ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_ADDFIELD( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_ADDFIELD( w, ip ) --> <see ( ( *( SUPERTABLE )->addField )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_APPEND( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_APPEND( w, l ) --> <see ( ( *( SUPERTABLE )->append )( w, l ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_CREATEFIELDS( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_CREATEFIELDS( w, v ) --> <see ( ( *( SUPERTABLE )->createFields )( w, v ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_DELETE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_DELETE( w ) --> <see ( ( *( SUPERTABLE )->deleterec )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_DELETED( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_DELETED( w, sp ) --> <see ( ( *( SUPERTABLE )->deleted )( w, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_FIELDCOUNT( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_FIELDCOUNT( w, sp ) --> <see ( ( *( SUPERTABLE )->fieldCount )( w, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_FIELDDISPLAY( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_FIELDDISPLAY( w, sp ) --> <see ( ( *( SUPERTABLE )->fieldDisplay )( w, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_FIELDINFO( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_FIELDINFO( w, s1, s2, v ) --> <see ( ( *( SUPERTABLE )->fieldInfo )( w, s1,
s2, v ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_FIELDNAME( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_FIELDNAME( w, i, bp ) --> <see ( ( *( SUPERTABLE )->fieldName )( w, i, bp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_FLUSH( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_FLUSH( w ) --> <see ( ( *( SUPERTABLE )->flush )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_GETREC( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_GETREC( w, bpp ) --> <see ( ( *( SUPERTABLE )->getRec )( w, bpp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_GETVALUE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_GETVALUE( w, i, v ) --> <see ( ( *( SUPERTABLE )->getValue )( w, i, v ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_GETVARLEN( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_GETVARLEN( w, i, lp ) --> <see ( ( *( SUPERTABLE )->getVarLen )( w, i, lp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_GOCOLD( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_GOCOLD( w ) --> <see ( ( *( SUPERTABLE )->goCold )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

**SUPER\_GOHOT( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_GOHOT( w ) --> <see ( ( *( SUPERTABLE )->goHot )( w ) )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapirdd.h

## Platforms

All

## **SUPER\_PUTVALUE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_PUTVALUE( w, i, v ) --> <see ( ( *( SUPERTABLE )->putValue )( w, i, v ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_PUTREC( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_PUTREC( w, bp ) --> <see ( ( *( SUPERTABLE )->putRec )( w, bp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_RECALL( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_RECALL( w ) --> <see ( ( *( SUPERTABLE )->recall )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_RECCOUNT( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_RECCOUNT( w, sp ) --> <see ( ( *( SUPERTABLE )->reccount )( w, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_RECINFO( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_RECINFO( w, v1, i, v2 ) --> <see ( ( *( SUPERTABLE )->recInfo )( w, v1, i, v2
) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

**SUPER\_RECNO( )**

## Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_RECNO( w, sp ) --> <see ( ( *( SUPERTABLE )->recno )( w, sp ) )>
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Header file is hbapirdd.h

## Platforms

All

## **SUPER\_SETFIELDEXTENT( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_SETFIELDEXTENT( w, s ) --> <see ( ( *( SUPERTABLE )->setFieldExtent )( w, s )
)>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_ALIAS( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_ALIAS( w, bp ) --> <see ( ( *( SUPERTABLE )->alias )( w, bp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_CLOSE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_CLOSE( w ) --> <see ( ( *( SUPERTABLE )->close )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_CREATE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_CREATE( w, ip ) --> <see ( ( *( SUPERTABLE )->create )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_INFO( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_INFO( w, i, g ) --> <see ( ( *( SUPERTABLE )->info )( w, i, g ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_NEW( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_NEW( w ) --> <see ( ( *( SUPERTABLE )->newarea )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_OPEN( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_OPEN( w, ip ) --> <see ( ( *( SUPERTABLE )->open )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_RELEASE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_RELEASE( w ) --> <see ( ( *( SUPERTABLE )->release )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_STRUCTSIZE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_STRUCTSIZE( w, sp ) --> <see ( ( *( SUPERTABLE )->structSize )( w, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_SYSNAME( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_SYSNAME( w, bp ) --> <see ( ( *( SUPERTABLE )->sysName )( w, bp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_DBEVAL( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_DBEVAL( w, ip ) --> <see ( ( *( SUPERTABLE )->dbEval )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_PACK( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_PACK( w ) --> <see ( ( *( SUPERTABLE )->pack )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_PACKREC( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_PACKREC( w, l, sp ) --> <see ( ( *( SUPERTABLE )->packRec )( w, l, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_SORT( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_SORT( w, ip ) --> <see ( ( *( SUPERTABLE )->sort )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_TRANS ( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_TRANS( w, ip ) --> <see ( ( *( SUPERTABLE )->trans )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_TRANSREC( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_TRANSREC( w, ip ) --> <see ( ( *( SUPERTABLE )->transRec )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_ZAP( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_ZAP( w ) --> <see ( ( *( SUPERTABLE )->zap )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_CHILDEND( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_CHILDEND( w, ip ) --> <see ( ( *( SUPERTABLE )->childEnd )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_CHILDSTART( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_CHILDSTART( w, ip ) --> <see ( ( *( SUPERTABLE )->childStart )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_CHILDSYNC( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_CHILDSYNC( w, ip ) --> <see ( ( *( SUPERTABLE )->childSync )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_SYNCCHILDREN( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_SYNCCHILDREN( w ) --> <see ( ( *( SUPERTABLE )->syncChildren )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_CLEARREL( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_CLEARREL( w ) --> <see ( ( *( SUPERTABLE )->clearRel )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_FORCEREL( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_FORCEREL( w ) --> <see ( ( *( SUPERTABLE )->forceRel )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_RELAREA( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_RELAREA( w, s, sp ) --> <see ( ( *( SUPERTABLE )->relArea )( w, s, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_RELEVAL( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_RELEVAL( w, ip ) --> <see ( ( *( SUPERTABLE )->relEval )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_RELTEXT( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_RELTEXT( w, s, bp ) --> <see ( ( *( SUPERTABLE )->relText )( w, s, bp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_SETREL( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_SETREL( w, ip ) --> <see ( ( *( SUPERTABLE )->setRel )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_ORDLSTADD( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_ORDLSTADD( w, lp ) --> <see ( ( *( SUPERTABLE )->orderListAdd )( w, lp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_ORDLSTDELETE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_ORDLSTDELETE( w, lp ) --> <see ( ( *( SUPERTABLE )->orderListDelete )( w, lp )
)>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SUPER\_ORDLSTFOCUS( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_ORDLSTFOCUS( w, lp ) --> <see ( ( *( SUPERTABLE )->orderListFocus )( w, lp )
)>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## **SUPER\_ORDLSTREBUILD( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_ORDLSTREBUILD( w ) --> <see ( ( *( SUPERTABLE )->orderListRebuild )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_ORDLSTCLEAR( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_ORDLSTCLEAR( w ) --> <see ( ( *( SUPERTABLE )->orderListClear )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_ORDSETCOND( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_ORDSETCOND( w, ip ) --> <see ( ( *( SUPERTABLE )->orderCondition )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_ORDCREATE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_ORDCREATE( w, ip ) --> <see ( ( *( SUPERTABLE )->orderCreate )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_ORDDELETE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_ORDDELETE( w, ip ) --> <see ( ( *( SUPERTABLE )->orderDelete )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_ORDINFO( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_ORDINFO( w, i, p ) --> <see ( ( *( SUPERTABLE )->orderInfo )( w, i, p ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SUPER\_ORDEXPR( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_ORDEXPR( w, p ) --> <see ( ( *( SUPERTABLE )->orderInfo )( w, DBOI_EXPRESSION,
p ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SUPER\_ORDCOND( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_ORDCOND( w, p ) --> <see ( ( *( SUPERTABLE )->orderInfo )( w, DBOI_CONDITION,
p ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SUPER\_ORDRECNO( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_ORDRECNO( w, p ) --> <see ( ( *( SUPERTABLE )->orderInfo )( w, DBOI_RECNO, p
) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SUPER\_ORDPOS ( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_ORDPOS( w, p ) --> <see ( ( *( SUPERTABLE )->orderInfo )( w, DBOI_POSITION,
p ) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SUPER\_ORDNUMBER( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_ORDNUMBER( w, p ) --> <see ( ( *( SUPERTABLE )->orderInfo )( w, DBOI_NUMBER, p
) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## SUPER\_ORDNAME( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_ORDNAME( w, p ) --> <see ( ( *( SUPERTABLE )->orderInfo )( w, DBOI_NAME, p )
)>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## **SUPER\_ORDBAGNAME( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_ORDBAGNAME( w, p ) --> <see ( ( *( SUPERTABLE )->orderInfo )( w,
DEOI_BAGNAME,p ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_ORDBAGEXT( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_ORDBAGEXT( w, p ) --> <see ( ( *( SUPERTABLE )->orderInfo )( w, DBOI_BAGEXT,
p ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_CLEARFILTER( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_CLEARFILTER( w ) --> <see ( ( *( SUPERTABLE )->clearFilter )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_CLEARLOCATE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_CLEARLOCATE( w ) --> <see ( ( *( SUPERTABLE )->clearLocate )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_CLEARSCOPE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_CLEARSCOPE( w ) --> <see ( ( *( SUPERTABLE )->clearScope )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SUPER\_COUNTSCOPE( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_COUNTSCOPE( w, ip, lp ) --> <see ( ( *( SUPERTABLE )->countScope )( w, ip, lp
) )>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

## **SUPER\_FILTERTEXT( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_FILTERTEXT( w, bp ) --> <see ( ( *( SUPERTABLE )->filterText )( w, bp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_SCOPEINFO( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_SCOPEINFO( w, i, v ) --> <see ( ( *( SUPERTABLE )->scopeInfo )( w, i, v ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_SETFILTER( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_SETFILTER( w, ip ) --> <see ( ( *( SUPERTABLE )->setFilter )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_SETLOCATE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_SETLOCATE( w, ip ) --> <see ( ( *( SUPERTABLE )->setLocate )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_SETSCOPE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_SETSCOPE( w, ip ) --> <see ( ( *( SUPERTABLE )->setScope )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_SKIPSCOPE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_SKIPSCOPE( w, bp, l ) --> <see ( ( *( SUPERTABLE )->skipScope )( w, bp, l ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_COMPILE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_COMPILE( w, bp ) --> <see ( ( *( SUPERTABLE )->compile )( w, bp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_ERROR( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_ERROR( w, ip ) --> <see ( ( *( SUPERTABLE )->error )( w, ip ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_EVALBLOCK( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_EVALBLOCK( w, v ) --> <see ( ( *( SUPERTABLE )->evalBlock )( w, v ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_GETLOCKS( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_GETLOCKS( w, g ) --> <see ( ( *( SUPERTABLE )->info )( w, DBI_GETLOCKARRAY, g
) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_RAWLOCK( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_RAWLOCK( w, i, l ) --> <see ( ( *( SUPERTABLE )->rawlock )( w, i, l ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_LOCK( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_LOCK( w, sp ) --> <see ( ( *( SUPERTABLE )->lock )( w, sp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_UNLOCK( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_UNLOCK( w ) --> <see ( ( *( SUPERTABLE )->unlock )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_CLOSEMEMFILE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_CLOSEMEMFILE( w ) --> <see ( ( *( SUPERTABLE )->closeMemFile )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_CREATEMEMFILE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_CREATEMEMFILE( w, bp ) --> <see ( ( *( SUPERTABLE )->createMemFile )( w, bp )
)>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_GETVALUEFILE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_GETVALUEFILE( w, i, bp ) --> <see ( ( *( SUPERTABLE )->getValueFile )( w, i,
bp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_OPENMEMFILE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_OPENMEMFILE( w, bp ) --> <see ( ( *( SUPERTABLE )->openMemFile )( w, bp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_PUTVALUEFILE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_PUTVALUEFILE( w, i, bp ) --> <see ( ( *( SUPERTABLE )->putValueFile )( w, i,
bp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_READDBHEADER( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_READDBHEADER( w ) --> <see ( ( *( SUPERTABLE )->readDBHeader )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_WRITEDBHEADER( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
```

```
SUPER_WRITEDBHEADER( w ) --> <see ( ( *( SUPERTABLE )->writeDBHeader )( w ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_RECSIZE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_RECSIZE( w, lp ) --> <see ( ( *( SUPERTABLE )->info )( w, DBI_GETRECSIZE, lp )
)>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_HEADERSIZE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_HEADERSIZE( w, fp ) --> <see ( ( *( SUPERTABLE )->info )( w,
DBI_GETHEADERSIZE, fp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_LUPDATE( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_LUPDATE( w, fp ) --> <see ( ( *( SUPERTABLE )->info )( w, DBI_LASTUPDATE, fp )
)>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_SETDELIM( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_SETDELIM( w, fp ) --> <see ( ( *( SUPERTABLE )->info )( w, DBI_SETDELIMITER,
fp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## **SUPER\_GETDELIM( )**

### **Syntax**

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_GETDELIM( w, fp ) --> <see ( ( *( SUPERTABLE )->info )( w, DBI_GETDELIMITER,
fp ) )>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Header file is hbapirdd.h

### **Platforms**

All

## SUPER\_TABLEEXT( )

### Syntax

C Prototype (macro definition)

```
#include <hbapirdd.h>
SUPER_TABLEEXT( w, fp ) --> <see ( ( *( SUPERTABLE )->info )( w, DBI_TABLEEXT, fp )
)>
```

### Arguments

### Returns

### Description

### Status

Ready  
Compliance is not applicable to API calls.

### Files

Header file is hbapirdd.h

### Platforms

All

**`_evalLaunch( )`**

## Syntax

C Prototype (macro replacement)

```
#include <item.api>
_evalLaunch --> <see hb_evalLaunch>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is item.api

## Platforms

All

## See Also:

[hb\\_evalLaunch\(\)](#)

**`_evalNew()`**

## Syntax

C Prototype (macro replacement)

```
#include <item.api>
_evalNew --> <see hb_evalNew>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is item.api

## Platforms

All

## See Also:

[hb\\_evalNew\(\)](#)

**`_evalPutParam( )`**

## Syntax

C Prototype (macro replacement)

```
#include <item.api>
_evalPutParam --> <see hb_evalPutParam>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is item.api

## Platforms

All

## See Also:

[hb\\_evalPutParam\(\)](#)

## **`_evalRelease()`**

### **Syntax**

C Prototype (macro replacement)

```
#include <item.api>
_evalRelease --> <see hb_evalRelease>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Header file is item.api

### **Platforms**

All

### **See Also:**

[hb\\_evalRelease\(\)](#)

**`_itemArrayGet()`**

## Syntax

C Prototype (macro replacement)

```
#include <item.api>
_itemArrayGet --> <see hb_itemArrayGet>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is item.api

## Platforms

All

## See Also:

[hb\\_itemArrayGet\(\)](#)

**`_itemArrayNew( )`**

## Syntax

C Prototype (macro replacement)

```
#include <item.api>
_itemArrayNew --> <see hb_itemArrayNew>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is item.api

## Platforms

All

## See Also:

[hb\\_itemArrayNew\(\)](#)

`_itemArrayPut()`

## Syntax

C Prototype (macro replacement)

```
#include <item.api>
_itemArrayPut --> <see hb_itemArrayPut>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is item.api

## Platforms

All

## See Also:

[hb\\_itemArrayPut\(\)](#)

**`_itemNew()`**

## Syntax

C Prototype (macro replacement)

```
#include <item.api>
_itemNew --> <see hb_itemNew>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is item.api

## Platforms

All

## See Also:

[`hb\_itemNew\(\)`](#)

**`_iParam()`**

## Syntax

C Prototype (macro replacement)

```
#include <item.api>
 iParam --> <see hb iParam>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is item.api

## Platforms

All

## See Also:

[hb iParam\(\)](#)

**`_itemRelease()`**

## Syntax

C Prototype (macro replacement)

```
#include <item.api>
_itemRelease --> <see hb_itemRelease>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is item.api

## Platforms

All

## See Also:

[hb\\_itemRelease\(\)](#)

**`_itemReturn()`**

## Syntax

C Prototype (macro replacement)

```
#include <item.api>
_itemReturn --> <see hb_itemReturn>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is item.api

## Platforms

All

## See Also:

[hb\\_itemReturn\(\)](#)

**`_itemSize()`**

## Syntax

C Prototype (macro replacement)

```
#include <item.api>
_itemSize --> <see hb_itemSize>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is item.api

## Platforms

All

## See Also:

[hb\\_itemSize\(\)](#)

**`_itemType()`**

## Syntax

C Prototype (macro replacement)

```
#include <item.api>
_itemType --> <see hb_itemType>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is item.api

## Platforms

All

## See Also:

[hb\\_itemType\(\)](#)

**`_reta()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_reta --> <see hb_reta>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_reta\(\)](#)

`_pcount ( )`

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_pcount --> <see hb_pcount>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_pcount\(\)](#)

**`_tchdir()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_tchdir --> <see hb_fsChDir>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_fsChDir\(\)](#)

**`_tchdrv()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_tchdrv --> <see hb_fsChDrv>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_fsChDrv\(\)](#)

**`_tclose()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_tclose --> <see hb_fsClose>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[`hb\_fsClose\(\)`](#)

`_tcommit()`

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_tcommit --> <see hb_fsCommit>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[`hb\_fsCommit\(\)`](#)

**`_tcreat()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_tcreat --> <see hb_fsCreate>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_fsCreate\(\)](#)

**`_tcurdir()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_tcurdir --> <see hb_fsCurDir>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_fsCurDir\(\)](#)

**`_tcurdrv()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_tcurdrv --> <see hb_fsCurDrv>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_fsCurDrv\(\)](#)

**`_tdevraw()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_tdevraw --> <see hb_fsSetDevRaw>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_fsSetDevRaw\(\)](#)

**`_terror()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_terror --> <see hb_fsError>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_fsError\(\)](#)

**`_tisdevice()`**

## Syntax

```
C Prototype (macro replacement)

#include <hbundoc.api>
_tisdevice --> <see hb_fsIsDevice>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_fsIsDevice\(\)](#)

**`_tisdrv()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_tisdrv --> <see hb_fsIsDrv>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[`hb\_fsIsDrv\(\)`](#)

**`_tlock()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_tlock --> <see hb_fsLock>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_fsLock\(\)](#)

**`_tlseek()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_tlseek --> <see hb_fsSeek>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_fsSeek\(\)](#)

**`_tmkdir()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_tmkdir --> <see hb_fsMkDir>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_fsMkDir\(\)](#)

**`_topen( )`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_topen --> <see hb_fsOpen>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_fsOpen\(\)](#)

**`_tread()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_tread --> <see hb_fsRead>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_fsRead\(\)](#)

**`_trename()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_trename --> <see hb_fsRename>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_fsRename\(\)](#)

**`_trmdir()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_trmdir --> <see hb_fsRmdir>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[`hb\_fsRmdir\(\)`](#)

**`_tunlink()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_tunlink --> <see hb_fsDelete>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[`hb\_fsDelete\(\)`](#)

**`_twrite()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_twrite --> <see hb_fsWrite>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_fsWrite\(\)](#)

**`_bset()`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_bset --> <see memset>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_parc\(\)](#)

**`_bmove( )`**

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_bmove --> <see memmove>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[`hb\_parcl\(\)`](#)

## `_bcopy( )`

### Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_bcopy --> <see memcpy>
```

### Arguments

### Returns

### Description

### Status

Ready

Header file is hbundoc.api

### Platforms

All

### See Also:

[hb\\_parcel\(\)](#)

`_bcmp( )`

## Syntax

C Prototype (macro replacement)

```
#include <hbundoc.api>
_bcmp --> <see memcmp>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is hbundoc.api

## Platforms

All

## See Also:

[hb\\_parcel\(\)](#)

**`_gtBox( )`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtBox --> <see hb_gtBox>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtBox\(\)](#)

`_gtColorSelect()`

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtColorSelect --> <see hb_gtColorSelect>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtColorSelect\(\)](#)

**`_gtDispBegin()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtDispBegin --> <see hb_gtDispBegin>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtDispBegin\(\)](#)

**`_gtDispCount()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtDispCount --> <see hb_gtDispCount>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtDispCount\(\)](#)

**`_gtDispEnd()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtDispEnd --> <see hb_gtDispEnd>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtDispEnd\(\)](#)

**`_gtGetColorStr()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtGetColorStr --> <see hb_gtGetColorStr>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtGetColorStr\(\)](#)

## `_gtGetCursor()`

### Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtGetCursor --> <see hb_gtGetCursor>
```

### Arguments

### Returns

### Description

### Status

Ready

Header file is gt.api

### Platforms

All

### See Also:

[hb\\_gtGetCursor\(\)](#)

**`_gtGetPos()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtGetPos --> <see hb_gtGetPos>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtGetPos\(\)](#)

**`_gtIsColor()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtIsColor --> <see hb_gtIsColor>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtIsColor\(\)](#)

**`_gtMaxCol()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtMaxCol --> <see hb_gtMaxCol>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtMaxCol\(\)](#)

**`_gtMaxRow( )`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtMaxRow --> <see hb_gtMaxRow>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtMaxRow\(\)](#)

**`_gtPostExt()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtPostExt --> <see hb_gtPostExt>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtPostExt\(\)](#)

**`_gtPreExt()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtPreExt --> <see hb_gtPreExt>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtPreExt\(\)](#)

**`_gtRectSize()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtRectSize --> <see hb_gtRectSize>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtRectSize\(\)](#)

**`_gtRepChar()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtRepChar --> <see hb_gtRepChar>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtRepChar\(\)](#)

**`_gtRest()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtRest --> <see hb_gtRest>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtRest\(\)](#)

**`_gtSave()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtSave --> <see hb_gtSave>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtSave\(\)](#)

`_gtScrDim()`

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtScrDim --> <see hb_gtScrDim>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtScrDim\(\)](#)

**`_gtScroll()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtScroll --> <see hb_gtScroll>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtScroll\(\)](#)

**`_gtSetBlink()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtSetBlink --> <see hb_gtSetBlink>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtSetBlink\(\)](#)

**`_gtSetColorStr()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtSetColorStr --> <see hb_gtSetColorStr>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtSetColorStr\(\)](#)

## `_gtSetCursor()`

### Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtSetCursor --> <see hb_gtSetCursor>
```

### Arguments

### Returns

### Description

### Status

Ready

Header file is gt.api

### Platforms

All

### See Also:

[hb\\_gtSetCursor\(\)](#)

**`_gtSetMode()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtSetMode --> <see hb_gtSetMode>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtSetMode\(\)](#)

**`_gtSetPos()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtSetPos --> <see hb_gtSetPos>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtSetPos\(\)](#)

**`_gtSetSnowFlag()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtSetSnowFlag --> <see hb_gtSetSnowFlag>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtSetSnowFlag\(\)](#)

**`_gtWrite()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtWrite --> <see hb_gtWrite>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtWrite\(\)](#)

## `_gtWriteAt()`

### Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtWriteAt --> <see hb_gtWriteAt>
```

### Arguments

### Returns

### Description

### Status

Ready

Header file is gt.api

### Platforms

All

### See Also:

[`hb\_gtWriteAt\(\)`](#)

**`_gtWriteCon()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtWriteCon --> <see hb_gtWriteCon>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtWriteCon\(\)](#)

**`_gtInit()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtInit --> <see hb_gtInit>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtInit\(\)](#)

**`_gtExit()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtExit --> <see hb_gtExit>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtExit\(\)](#)

**`_gtWCreate()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtWCreate --> <see hb_gtWCreate>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtWCreate\(\)](#)

**`_gtWDestroy()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtWDestroy --> <see hb_gtWDestroy>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtWDestroy\(\)](#)

**`_gtWFlash()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtWFlash --> <see hb_gtWFlash>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtWFlash\(\)](#)

**`_gtWApp( )`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtWApp --> <see hb_gtWApp>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtWApp\(\)](#)

**`_gtWCurrent()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtWCurrent --> <see hb_gtWCurrent>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtWCurrent\(\)](#)

**`_gtWPos()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtWPos --> <see hb_gtWPos>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtWPos\(\)](#)

**`_gtWVis()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtWVis --> <see hb_gtWVis>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtWVis\(\)](#)

**`_gtModalRead()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtModalRead --> <see hb_gtModalRead>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtModalRead\(\)](#)

**`_gtBeginWrite()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtBeginWrite --> <see hb_gtBeginWrite>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtBeginWrite\(\)](#)

**`_gtEndWrite()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtEndWrite --> <see hb_gtEndWrite>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtEndWrite\(\)](#)

**`_gtFlushCursor()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtFlushCursor --> <see hb_gtFlushCursor>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtFlushCursor\(\)](#)

**`_gtSetColor()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtSetColor --> <see hb_gtSetColor>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtSetColor\(\)](#)

**`_gtGetColor()`**

## Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtGetColor --> <see hb_gtGetColor>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is gt.api

## Platforms

All

## See Also:

[hb\\_gtGetColor\(\)](#)

## `_gtSetBorder()`

### Syntax

C Prototype (macro replacement)

```
#include <gt.api>
_gtSetBorder --> <see hb_gtSetBorder>
```

### Arguments

### Returns

### Description

### Status

Ready

Header file is gt.api

### Platforms

All

### See Also:

[hb\\_gtSetBorder\(\)](#)

**`_xalloc()`**

## Syntax

C Prototype (macro replacement)

```
#include <fm.api>
_xalloc --> <see hb_xalloc>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is fm.api

## Platforms

All

## See Also:

[`hb\_xalloc\(\)`](#)

**`_xgrab( )`**

## Syntax

C Prototype (macro replacement)

```
#include <fm.api>
_xgrab --> <see hb_xgrab>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is fm.api

## Platforms

All

## See Also:

[hb\\_xgrab\(\)](#)

## **`_xfree()`**

### **Syntax**

C Prototype (macro replacement)

```
#include <fm.api>
_xfree --> <see hb_xfree>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Header file is fm.api

### **Platforms**

All

### **See Also:**

[hb\\_xfree\(\)](#)

**`_exmgrab()`**

## Syntax

C Prototype (macro replacement)

```
#include <fm.api>
_exmgrab --> <see hb_xgrab>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is fm.api

## Platforms

All

## See Also:

[hb\\_xgrab\(\)](#)

**`_fsChDir()`**

## Syntax

C Prototype (macro replacement)

```
#include <fileysys.api>
_fsChDir --> <see hb_fsChDir>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is fileysys.api

## Platforms

All

## See Also:

[hb\\_fsChDir\(\)](#)

**`_fsChDrv()`**

## Syntax

C Prototype (macro replacement)

```
#include <filesys.api>
_fsChDrv --> <see hb_fsChDrv>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is filesys.api

## Platforms

All

## See Also:

[hb\\_fsChDrv\(\)](#)

**`_fsClose()`**

## Syntax

C Prototype (macro replacement)

```
#include <fileysys.api>
_fsClose --> <see hb_fsClose>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is fileysys.api

## Platforms

All

## See Also:

[hb\\_fsClose\(\)](#)

**`_fsCommit()`**

## Syntax

C Prototype (macro replacement)

```
#include <filesystem.api>
_fsCommit --> <see hb_fsCommit>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is filesystem.api

## Platforms

All

## See Also:

[`hb\_fsCommit\(\)`](#)

**`_fsCreate()`**

## Syntax

C Prototype (macro replacement)

```
#include <filesystem.api>
_fsCreate --> <see hb_fsCreate>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is filesystem.api

## Platforms

All

## See Also:

[hb\\_fsCreate\(\)](#)

**`_fsCurDir()`**

## Syntax

C Prototype (macro replacement)

```
#include <fileysys.api>
_fsCurDir --> <see hb_fsCurDir>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is fileysys.api

## Platforms

All

## See Also:

[hb\\_fsCurDir\(\)](#)

**`_fsCurDrv( )`**

## Syntax

C Prototype (macro replacement)

```
#include <fileSYS.api>
_fsCurDrv --> <see hb_fsCurDrv>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is fileSYS.api

## Platforms

All

## See Also:

[hb\\_fsCurDrv\(\)](#)

**`_fsDelete()`**

## Syntax

C Prototype (macro replacement)

```
#include <filesystem.api>
_fsDelete --> <see hb_fsDelete>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is filesystem.api

## Platforms

All

## See Also:

[hb\\_fsDelete\(\)](#)

**`_fsError()`**

## Syntax

C Prototype (macro replacement)

```
#include <fileSYS.api>
_fsError --> <see hb_fsError>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is fileSYS.api

## Platforms

All

## See Also:

[hb\\_fsError\(\)](#)

## **`_fsExtOpen()`**

### **Syntax**

```
C Prototype (macro replacement)

#include <fileysys.api>
_fsExtOpen --> <see hb_fsExtOpen>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Header file is fileysys.api

### **Platforms**

All

### **See Also:**

[hb\\_fsExtOpen\(\)](#)

**`_fsIsDrv()`**

## Syntax

C Prototype (macro replacement)

```
#include <filesys.api>
_fsIsDrv --> <see hb_fsIsDrv>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is filesys.api

## Platforms

All

## See Also:

[hb\\_fsIsDrv\(\)](#)

**`_fsLock()`**

## Syntax

C Prototype (macro replacement)

```
#include <filesystem.api>
_fsLock --> <see hb_fsLock>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is filesystem.api

## Platforms

All

## See Also:

[hb\\_fsLock\(\)](#)

**`_fsMkDir()`**

## Syntax

C Prototype (macro replacement)

```
#include <fileysys.api>
_fsMkDir --> <see hb_fsMkDir>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is fileysys.api

## Platforms

All

## See Also:

[hb\\_fsMkDir\(\)](#)

**`_fsOpen()`**

## Syntax

C Prototype (macro replacement)

```
#include <fileysys.api>
_fsOpen --> <see hb_fsOpen>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is fileysys.api

## Platforms

All

## See Also:

[hb\\_fsOpen\(\)](#)

**`_fsRead()`**

## Syntax

C Prototype (macro replacement)

```
#include <filesys.api>
_fsRead --> <see hb_fsRead>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is filesys.api

## Platforms

All

## See Also:

[hb\\_fsRead\(\)](#)

**`_fsRmdir()`**

## Syntax

C Prototype (macro replacement)

```
#include <fileysys.api>
_fsRmdir --> <see hb_fsRmdir>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is fileysys.api

## Platforms

All

## See Also:

[hb\\_fsRmdir\(\)](#)

**`_fsRename()`**

## Syntax

C Prototype (macro replacement)

```
#include <filesystem.api>
_fsRename --> <see hb_fsRename>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is filesystem.api

## Platforms

All

## See Also:

[hb\\_fsRename\(\)](#)

## **`_fsSeek()`**

### **Syntax**

C Prototype (macro replacement)

```
#include <fileysys.api>
_fsSeek --> <see hb_fsSeek>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Header file is fileysys.api

### **Platforms**

All

### **See Also:**

[hb\\_fsSeek\(\)](#)

## **`_fsWrite()`**

### **Syntax**

C Prototype (macro replacement)

```
#include <fileSYS.api>
_fsWrite --> <see hb_fsWrite>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Header file is fileSYS.api

### **Platforms**

All

### **See Also:**

[hb\\_fsWrite\(\)](#)

# ALENGTH

## Syntax

C Prototype (macro definition)

```
#include <extend.api>
ALENGTH( n ) --> <see hb_parinfa( n, 0 )>
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Header file is extend.api

## Platforms

All

`_parc()`

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_parc --> <see hb_parc>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_parc\(\)](#)

**`_parclen()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_parclen --> <see hb_parclen>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[`hb\_parclen\(\)`](#)

**`_parcsiz()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_parcsiz --> <see hb_parcsiz>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[`hb\_parcsiz\(\)`](#)

**`_pards( )`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_pards --> <see hb_pards>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_pards\(\)](#)

**`_parinfo()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_parinfo --> <see hb_parinfo>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[`hb\_parinfo\(\)`](#)

**`_parinfo()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_parinfo --> <see hb_parinfo>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[`hb\_parinfo\(\)`](#)

`_parl()`

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_parl --> <see hb_parl>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_parl\(\)](#)

**`_parnd( )`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_parnd --> <see hb_parnd>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_parnd\(\)](#)

`_parni()`

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_parni --> <see hb_parni>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_parni\(\)](#)

`_parnl()`

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_parnl --> <see hb_parnl>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_parnl\(\)](#)

**`_ret()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_ret --> <see hb_ret>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_ret\(\)](#)

**`_retc()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_retc --> <see hb_retch>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_retch\(\)](#)

**`_retclen()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_retclen --> <see hb_retclen>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[`hb\_retclen\(\)`](#)

**`_retds()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_retlds --> <see hb_retlds>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_retlds\(\)](#)

**`_retl()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_ret1 --> <see hb_ret1>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_ret1\(\)](#)

**`_retnd()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_retnd --> <see hb_retnd>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_retnd\(\)](#)

**`_retni()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_retni --> <see hb_retni>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_retni\(\)](#)

**`_retnl()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_retnl --> <see hb_retnl>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_retnl\(\)](#)

**`_storc()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_storc --> <see hb_storc>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_storc\(\)](#)

**`_storclen()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_storclen --> <see hb_storclen>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_storclen\(\)](#)

**`_stords()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_stords --> <see hb_stords>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_stords\(\)](#)

**`_storl()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_storl --> <see hb_storl>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_storl\(\)](#)

**`_stornd()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_stornd --> <see hb_stornd>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_stornd\(\)](#)

**`_storni()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_storni --> <see hb_storni>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_storni\(\)](#)

**`_stornl()`**

## Syntax

C Prototype (macro replacement)

```
#include <extend.api>
_stornl --> <see hb_stornl>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is extend.api

## Platforms

All

## See Also:

[hb\\_stornl\(\)](#)

## `_errGetDescription()`

### Syntax

C Prototype (macro replacement)

```
#include <error.api>
```

```
_errGetDescription --> <see hb_errGetDescription>
```

### Arguments

### Returns

### Description

### Status

Ready

Header file is error.api

### Platforms

All

### See Also:

[hb\\_errGetDescription\(\)](#)

**`_errGetFileName()`**

## Syntax

C Prototype (macro replacement)

```
#include <error.api>
_errGetFileName --> <see hb_errGetFileName>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is error.api

## Platforms

All

## See Also:

[hb\\_errGetFileName\(\)](#)

## **`_errGetFlags()`**

### **Syntax**

C Prototype (macro replacement)

```
#include <error.api>
_errGetFlags --> <see hb_errGetFlags>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Header file is error.api

### **Platforms**

All

### **See Also:**

[hb\\_errGetFlags\(\)](#)

## **`_errGetGenCode()`**

### **Syntax**

C Prototype (macro replacement)

```
#include <error.api>
_errGetGenCode --> <see hb_errGetGenCode>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Header file is error.api

### **Platforms**

All

### **See Also:**

[hb\\_errGetGenCode\(\)](#)

## `_errGetOperation()`

### Syntax

C Prototype (macro replacement)

```
#include <error.api>
_errGetOperation --> <see hb_errGetOperation>
```

### Arguments

### Returns

### Description

### Status

Ready

Header file is error.api

### Platforms

All

### See Also:

[hb\\_errGetOperation\(\)](#)

**`_errGetOsCode( )`**

## Syntax

C Prototype (macro replacement)

```
#include <error.api>
_errGetOsCode --> <see hb_errGetOsCode>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is error.api

## Platforms

All

## See Also:

[hb\\_errGetOsCode\(\)](#)

## `_errGetSeverity()`

### Syntax

C Prototype (macro replacement)

```
#include <error.api>
_errGetSeverity --> <see hb_errGetSeverity>
```

### Arguments

### Returns

### Description

### Status

Ready

Header file is error.api

### Platforms

All

### See Also:

[hb\\_errGetSeverity\(\)](#)

**`_errGetSubCode()`**

## Syntax

C Prototype (macro replacement)

```
#include <error.api>
_errGetSubCode --> <see hb_errGetSubCode>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is error.api

## Platforms

All

## See Also:

[hb\\_errGetSubCode\(\)](#)

**`_errGetSubSystem()`**

## Syntax

C Prototype (macro replacement)

```
#include <error.api>
_errGetSubSystem --> <see hb_errGetSubSystem>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is error.api

## Platforms

All

## See Also:

[hb\\_errGetSubSystem\(\)](#)

## **`_errGetTries()`**

### **Syntax**

C Prototype (macro replacement)

```
#include <error.api>
_errGetTries --> <see hb_errGetTries>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Header file is error.api

### **Platforms**

All

### **See Also:**

[hb\\_errGetTries\(\)](#)

## **`_errLaunch()`**

### **Syntax**

```
C Prototype (macro replacement)

#include <error.api>
_errLaunch --> <see hb_errLaunch>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Header file is error.api

### **Platforms**

All

### **See Also:**

[hb\\_errLaunch\(\)](#)

**`_errNew( )`**

## Syntax

C Prototype (macro replacement)

```
#include <error.api>
_errNew --> <see hb_errNew>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is error.api

## Platforms

All

## See Also:

[hb\\_errNew\(\)](#)

## `_errPutDescription()`

### Syntax

C Prototype (macro replacement)

```
#include <error.api>
_errPutDescription --> <see hb_errPutDescription>
```

### Arguments

### Returns

### Description

### Status

Ready

Header file is error.api

### Platforms

All

### See Also:

[hb\\_errPutDescription\(\)](#)

**`_errPutFileName( )`**

## Syntax

C Prototype (macro replacement)

```
#include <error.api>
_errPutFileName --> <see hb_errPutFileName>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is error.api

## Platforms

All

## See Also:

[hb\\_errPutFileName\(\)](#)

## **`_errPutFlags()`**

### **Syntax**

C Prototype (macro replacement)

```
#include <error.api>
_errPutFlags --> <see hb_errPutFlags>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Header file is error.api

### **Platforms**

All

### **See Also:**

[hb\\_errPutFlags\(\)](#)

## **`_errPutGenCode()`**

### **Syntax**

C Prototype (macro replacement)

```
#include <error.api>
_errPutGenCode --> <see hb_errPutGenCode>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Header file is error.api

### **Platforms**

All

### **See Also:**

[hb\\_errPutGenCode\(\)](#)

## `_errPutOperation()`

### Syntax

C Prototype (macro replacement)

```
#include <error.api>
_errPutOperation --> <see hb_errPutOperation>
```

### Arguments

### Returns

### Description

### Status

Ready

Header file is error.api

### Platforms

All

### See Also:

[`hb\_errPutOperation\(\)`](#)

**`_errPutOsCode( )`**

## Syntax

C Prototype (macro replacement)

```
#include <error.api>
_errPutOsCode --> <see hb_errPutOsCode>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is error.api

## Platforms

All

## See Also:

[hb\\_errPutOsCode\(\)](#)

# `_errPutSeverity()`

## Syntax

C Prototype (macro replacement)

```
#include <error.api>
_errPutSeverity --> <see hb_errPutSeverity>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is error.api

## Platforms

All

## See Also:

[hb\\_errPutSeverity\(\)](#)

**`_errPutSubCode( )`**

## Syntax

C Prototype (macro replacement)

```
#include <error.api>
_errPutSubCode --> <see hb_errPutSubCode>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is error.api

## Platforms

All

## See Also:

[hb\\_errPutSubCode\(\)](#)

**`_errPutSubSystem()`**

## Syntax

C Prototype (macro replacement)

```
#include <error.api>
_errPutSubSystem --> <see hb_errPutSubSystem>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is error.api

## Platforms

All

## See Also:

[hb\\_errPutSubSystem\(\)](#)

## **`_errPutTries()`**

### **Syntax**

C Prototype (macro replacement)

```
#include <error.api>
_errPutTries --> <see hb_errPutTries>
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Header file is error.api

### **Platforms**

All

### **See Also:**

[hb\\_errPutTries\(\)](#)

**`_errRelease()`**

## Syntax

C Prototype (macro replacement)

```
#include <error.api>
_errRelease --> <see hb_errRelease>
```

## Arguments

## Returns

## Description

## Status

Ready

Header file is error.api

## Platforms

All

## See Also:

[hb\\_errRelease\(\)](#)

**hb\_dateSeconds( )**

## Syntax

C Prototype

```
#include <hbdate.h>
hb_dateSeconds( void ) --> ( double )dResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_dateToday( )**

## Syntax

C Prototype

```
#include <hbdate.h>
hb_dateToday( long * pYear, long * pMonth, long * pDay ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_dateTimeStr()**

## Syntax

C Prototype

```
#include <hbdate.h>
hb_dateTimeStr( char * pszTime ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_dateCMonth( )**

## Syntax

C Prototype

```
#include <hbdate.h>
hb_dateCMonth( int iMonth ) --> ( char * )pszResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_dateCDOW( )**

## Syntax

C Prototype

```
#include <hbdate.h>
hb_dateCDOW( int iDay ) --> ( char * )pszResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_dateDOW( )**

## Syntax

C Prototype

```
#include <hbdate.h>
hb_dateDOW( long lYear, long lMonth, long lDay ) --> ( long )lResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_dateFormat()**

## Syntax

C Prototype

```
#include <hbdate.h>
hb_dateFormat( const char * szDate, char * szFormattedDate, const char *
szDateFormat ) --> ( char * )pszResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_dateEncode( )**

## Syntax

C Prototype

```
#include <hbdate.h>
hb_dateEncode( long lYear, long lMonth, long lDay ) --> ( long )lResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_dateDecode( )**

## Syntax

C Prototype

```
#include <hbdate.h>
hb_dateDecode( long julian, long * pYear, long * pMonth, long * pDay ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_dateStrPut( )**

## Syntax

C Prototype

```
#include <hbdate.h>
hb_dateStrPut( char * szDate, long lYear, long lMonth, long lDay ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

# hb\_dateStrGet( )

## Syntax

C Prototype

```
#include <hbdate.h>
hb_dateStrGet( const char * szDate, long * plYear, long * plMonth, long * plDay )
--> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_dateDecStr( )**

## Syntax

C Prototype

```
#include <hbdate.h>
hb_dateDecStr( char * szDate, long lJulian ) --> ( char * )pszResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_dateEncStr( )**

## Syntax

C Prototype

```
#include <hbdate.h>
hb_dateEncStr( char * szDate ) --> ( long )lResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_macroError( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_macroError( int iError, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_macroYYParse( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_macroYYParse( HB_MACRO_PTR pMacro ) --> ( int )iResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenPCode1( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenPCode1( BYTE byte, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenPCode2( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenPCode2( BYTE byte1, BYTE byte2, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenPCode3( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
```

```
hb_compGenPCode3( BYTE byte1, BYTE byte2, BYTE byte3, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenPCode4( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenPCode4( BYTE byte1, BYTE byte2, BYTE byte3, BYTE byte4, HB_BISON_PTR
pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenPCodeN( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenPCodeN( BYTE * pBuffer, ULONG ulSize, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compLocalVarGetPos( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compLocalVarGetPos( char * szVarName, HB_BISON_PTR pMacro ) --> ( int )iResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenJump( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenJump( LONG lOffset, HB_BISON_PTR pMacro ) --> ( ULONG )ulResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenJumpFalse( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
```

```
hb_compGenJumpFalse( LONG lOffset, HB_BISON_PTR pMacro ) --> ( ULONG )ulResult
```

## Arguments

## Returns

## Description

## Status

Ready

Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenJumpThere( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenJumpThere( ULONG ulFrom, ULONG ulTo, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenJumpHere( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenJumpHere( ULONG ulOffset, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenJumpTrue( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenJumpTrue( LONG lOffset, HB_BISON_PTR pMacro ) --> ( ULONG )ulResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compMemvarGenPCode( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compMemvarGenPCode( BYTE bPCode, char * szVarName, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenPushSymbol()**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenPushSymbol( char * szSymbolName, int isFunction, HB_BISON_PTR pMacro ) -->
void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenPushLong( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenPushLong( long lNumber, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenMessage( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenMessage( char * szMsgName, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenMessageData( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenMessageData( char * szMsg, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenPopVar( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenPopVar( char * szVarName, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenPopAliasedVar( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenPopAliasedVar( char * szVarName, BOOL bPushAliasValue, char * szAlias,
long lWorkarea, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenPushVar( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenPushVar( char * szVarName, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compGenPushVarRef( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenPushVarRef( char * szVarName, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

## **hb\_compGenPushAliasedVar()**

### **Syntax**

C Prototype

```
#include <hbmacro.h>
hb_compGenPushAliasedVar( char * szVarName, BOOL bPushAliasValue, char * szAlias,
long lWorkarea, HB_BISON_PTR pMacro ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is macro

### **Platforms**

All

**hb\_compGenPushLogical()**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenPushLogical( int iTrueFalse, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

## **hb\_compGenPushDouble()**

### **Syntax**

C Prototype

```
#include <hbmacro.h>
hb_compGenPushDouble( double dNumber, BYTE bWidth, BYTE bDec, HB_BISON_PTR pMacro )
--> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is macro

### **Platforms**

All

**hb\_compGenPushFunCall()**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compGenPushFunCall( char * szFunName, HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

## **hb\_compGenPushString()**

### **Syntax**

C Prototype

```
#include <hbmacro.h>
hb_compGenPushString( char * szText, ULONG ulStrLen, HB_BISON_PTR pMacro ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is macro

### **Platforms**

All

**hb\_compCodeBlockStart( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compCodeBlockStart( HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

**hb\_compCodeBlockEnd( )**

## Syntax

C Prototype

```
#include <hbmacro.h>
hb_compCodeBlockEnd( HB_BISON_PTR pMacro ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is macro

## Platforms

All

# hb\_setInitialize()

## Syntax

C Prototype

```
#include <hbset.h>
hb_setInitialize( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

**hb\_setRelease( )**

## Syntax

C Prototype

```
#include <hbset.h>
hb_setRelease( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## hb\_setListenerAdd( )

### Syntax

C Prototype

```
#include <hbset.h>
hb_setListenerAdd( PHB_SET_LISTENER_CALLBACK callback ) --> int
```

### Arguments

value. The first parameter identifies the SET parameter that is to be changed and the second parameter identifies whether the call is from before or after the value is changed. The callback function will be called twice whenever a SET parameter is changed using the Harbour SET function. The first call takes place before the SET value is changed and the second one is after the SET parameter has been changed.

### Returns

deactivate the callback function.

### Description

This function allows a subsystem that needs to track the status of some SET parameters to be notified whenever a SET parameter gets changed.

### Examples

```
void callback_function( HB_set_enum set, HB_set_listener_enum when )
{
    printf("\nCalled for SET parameter %d %s changing.",
        set, (when ? "after" : "before"));
}
int handle = hb_setListenerAdd( callback_function );
```

### Status

Ready

### Compliance

Compliance is not applicable to API calls.

### Files

Library is rtl

### Platforms

All

### See Also:

[hb\\_setListenerRemove\(\)](#)

# hb\_setListenerNotify()

## Syntax

C Prototype

```
#include <hbset.h>
hb_setListenerNotify( HB_set_enum set, HB_set_listener_enum
when ) --> int
```

## Arguments

HB\_SET\_LISTENER\_BEFORE when called before the SET parameter is to be changed and  
set to HB\_SET\_LISTENER\_AFTER when called after the SET parameter has been changed.

## Returns

## Description

This function notifies all SET listener callback functions. It must be called any time you change the value of a SET parameter directly instead of using the Harbour SET function. Both before and after the change.

## Examples

```
hb_setListenerNotify( HB_SET_DECIMALS, HB_SET_LISTENER_BEFORE );
hb_set.HB_SET_DECIMALS = 3;
hb_setListenerNotify( HB_SET_DECIMALS, HB_SET_LISTENER_AFTER );
```

## Status

Ready

## Compliance

Compliance is not applicable to API calls.

## Files

Library is rtl

## Platforms

All

## See Also:

[hb\\_setListenerAdd\(\)](#)

## **hb\_setListenerRemove()**

### **Syntax**

C Prototype

```
#include <hbset.h>
hb_setListenerRemove( int handle ) --> int
```

### **Arguments**

### **Returns**

the handle if the callback function was removed.

### **Description**

This function removes a SET listener callback function.

### **Examples**

```
int handle = hb_setListenerAdd( callback_function );
...
hb_setListenerRemove( handle );
```

### **Status**

Ready

### **Compliance**

Compliance is not applicable to API calls.

### **Files**

Library is rtl

### **Platforms**

All

### **See Also:**

[hb\\_setListenerAdd\(\)](#)

**hb\_vmInit()**

## Syntax

C Prototype

```
#include <hbvm.h>
hb_vmInit( BOOL bStartMainProc ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

## **hb\_vmQuit()**

Immediately quits the virtual machine

### **Syntax**

C Prototype

```
#include <hbvm.h>
hb_vmQuit( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmExecute()**

Invokes the virtual machine

### **Syntax**

C Prototype

```
#include <hbvm.h>
hb_vmExecute( const BYTE * pCode, PHB_SYMB pSymbols ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

**hb\_vmProcessSymbols()**  
Statics symbols initialization

## Syntax

C Prototype

```
#include <hbvm.h>
hb_vmProcessSymbols( PHB_SYMB pSymbols, USHORT uiSymbols ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

## **hb\_vmSymbolInit\_RT()**

Initialization of runtime support symbols

### **Syntax**

C Prototype

```
#include <hbvm.h>
hb_vmSymbolInit_RT( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

**hb\_vmRequestQuit()**

## Syntax

C Prototype

```
#include <hbvm.h>
hb_vmRequestQuit( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

**hb\_vmRequestEndProc( )**

## Syntax

C Prototype

```
#include <hbvm.h>
hb_vmRequestEndProc( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

**hb\_vmRequestCancel()**

## Syntax

C Prototype

```
#include <hbvm.h>
hb_vmRequestCancel( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

**hb\_vmRequestBreak( )**

## Syntax

C Prototype

```
#include <hbvm.h>
hb_vmRequestBreak( PHB_ITEM pItem ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

**hb\_vmRequestQuery( )**

## Syntax

C Prototype

```
#include <hbvm.h>
hb_vmRequestQuery( void ) --> ( USHORT )usResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

## **hb\_vmMessage( )**

Sends a message to an object

### **Syntax**

C Prototype

```
#include <hbvm.h>
hb_vmMessage( PHB_SYMB pSymMsg ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmDo( )**

Invoke the virtual machine

### **Syntax**

C Prototype

```
#include <hbvm.h>
hb_vmDo( USHORT uiParams ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmFunction()**

Executes a function saving its result

### **Syntax**

C Prototype

```
#include <hbvm.h>
hb_vmFunction( USHORT uiParams ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmSend( )**

Sends a message to an object

### **Syntax**

C Prototype

```
#include <hbvm.h>
hb_vmSend( USHORT uiParams ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmEvalBlock()**

Executes passed codeblock with no arguments

### **Syntax**

C Prototype

```
#include <hbvm.h>
```

```
hb_vmEvalBlock( PHB_ITEM pBlockItem ) --> ( PHB_ITEM )pResult
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

**hb\_vmEvalBlockV()**

## Syntax

C Prototype

```
#include <hbvm.h>
hb_vmEvalBlockV( PHB_ITEM pBlockItem, USHORT uiArgCount, ... ) --> ( PHB_ITEM
)pResult
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

## **hb\_vmPush( )**

Pushes a generic item onto the stack

### **Syntax**

C Prototype

```
#include <hbvm.h>
hb_vmPush( PHB_ITEM pItem ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmPushNil()**

In this case it places nil at self

### **Syntax**

C Prototype

```
#include <hbvm.h>
hb_vmPushNil( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmPushNumber( )**

Pushes a number on to the stack and decides if it is integer, long or double

### **Syntax**

C Prototype

```
#include <hbvm.h>
```

```
hb_vmPushNumber( double dNumber, int iDec ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmPushInteger()**

Pushes a integer number onto the stack

### **Syntax**

C Prototype

```
#include <hbvm.h>
hb_vmPushInteger( int iNumber ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmPushLong( )**

Pushes a long number onto the stack

### **Syntax**

C Prototype

```
#include <hbvm.h>
hb_vmPushLong( long lNumber ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmPushDouble()**

Pushes a double number onto the stack

### **Syntax**

C Prototype

```
#include <hbvm.h>
```

```
hb_vmPushDouble( double lNumber, int iDec ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmPushLogical()**

Pushes a logical value onto the stack

### **Syntax**

C Prototype

```
#include <hbvm.h>
hb_vmPushLogical( BOOL bValue ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmPushString()**

Pushes a string on to the stack

### **Syntax**

C Prototype

```
#include <hbvm.h>
```

```
hb_vmPushString( char * szText, ULONG length ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmPushDate()**

Pushes a long date onto the stack

### **Syntax**

C Prototype

```
#include <hbvm.h>
hb_vmPushDate( long lDate ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmPushSymbol()**

Pushes a function pointer onto the stack

### **Syntax**

C Prototype

```
#include <hbvm.h>
hb_vmPushSymbol( PHB_SYMB pSym ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready  
Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

## **hb\_vmPushPointer()**

Push an item of HB\_IT\_POINTER type

### **Syntax**

C Prototype

```
#include <hbvm.h>
hb_vmPushPointer( void * ) --> void
```

### **Arguments**

<void \*>

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All

**hb\_stackDispCall()**

## Syntax

C Prototype

```
#include <hbvm.h>
hb_stackDispCall( void ) --> void
```

## Arguments

## Returns

## Description

## Status

Ready  
Compliance is not applicable to API calls.

## Files

Library is vm

## Platforms

All

## **hb\_stackPop()**

Pops an item from the stack

### **Syntax**

C Prototype

```
#include <hbvm.h>
hb_stackPop( void ) --> void
```

### **Arguments**

### **Returns**

### **Description**

### **Status**

Ready

Compliance is not applicable to API calls.

### **Files**

Library is vm

### **Platforms**

All