

The intData Structure

```
struct _far intData
    char  shMonths[12][20];
    char  loMonths[12][20];
    char  veShDays[7][20];
    char  shDays[7][20];
    char  loDays[7][20];
    BOOL  shortMonth ;
    BOOL  shortDay;
    BOOL  shortYear;
    int   longDateMonthFormat;
    int   longDateYearFormat;
    int   longDateWeekDayFormat;
    int   longDateDayFormat;
    int   longDateSequence;
    char  longDateFirstSeparator[6];
    char  longDateSecondSeparator[6];
    char  longDateThirdSeparator[6];
    int   dateFormat;
    char  dateSeparator;

    int   iTime;
    BOOL  iTLZero;
    char  sTime[2];
    char  am[7];
    char  pm[7];

    int   iCurrency;
    int   iCurrDigits;
    char  SZCurrency[5];
    int   iNegCurr;
    int   iLZero;

    int   iDigits;
    int   iMeasure;
    char  iThousand[2];
    char  iDecimal[2];
    int  language;
;
```

REMEMBER: ANY CHANGE TO THIS STRUCTURE WILL AFFECT ALL THE APPLICATIONS

```
char  shMonths[12]
```

Short months names as Jan. for January

char loMonths[12]

Long Months names as January, February etc..

char veShDays[7]

Very Short Days as M for Monday

char shDays[7]

Short Days as Mon. For Monday

char loDays[7][20]

Long Days as Monday, Saturday ecc..

BOOL shortMonth

TRUE if in the short date format is the day is represented as 2 FALSE if it is represented as 02

BOOL shortDay

TRUE if in the short date format the day is represented as 2 FALSE if it is represented as 02

BOOL shortYear

TRUE if in the short date format the year is represented as 92 FALSE if it is represented as 1992

int dateFormat

0 if the long date is represented as MDY:

1 if the long date is represented as DMY:

2 if the long date is represented as YMD:

char dateSeparator

Separator between the short date elements

int longDateMonthFormat

1 if the month in the long date format is represented as 3

2 if the month in the long date format is represented as 03

3 if the month in the long date format is represented in a Short Month format Mar.

4 if the month in the long date format is represented in a Long Month format March.

int longDateWeekDayFormat

1 if the week day in the long date format is represented as short week day (Mon.)

2 if the week day in the long date format is represented as long week day (Mon.)

int longDateDayFormat

1 if the day in the long date format is represented as 3

2 if the day in the long date format is represented as 03

int longDateYearFormat

2 if the year in the long date format is represented as 92
4 if the year in the long date format is represented as 1992

int longDateSequence

0 if the long date is represented as MDY:
1 if the long date is represented as DMY:
2 if the long date is represented as YMD:

char longDateFirstSeparator[6]

char longDateSecondSeparator[6]

char longDateThirdSeparator[6]

First Second and Third long date separators.

int iTime

0 if the time is represented on a 24 hours base.
1 if the time is represented on a 12 hours base.

BOOL iTLZero

TRUE if in the short date format the year is represented as 92 FALSE if it is represented as 1992

char sTime[2]

Separator between Hours, Minutes, Seconds in a time.

char am[7]

char pm[7]

Strings used to signal an Anti or Post Meridian Time.

int iNegCurr

0 if a negative currency is represented as (\$12,2)
1 if a negative currency is represented as -\$12,22
2 if a negative currency is represented as \$-12,22
3 if a negative currency is represented as \$12,22-
4 if a negative currency is represented as (12,22\$)
5 if a negative currency is represented as -12,22\$
6 if a negative currency is represented as 12,22-\$
7 if a negative currency is represented as 12,22\$-
8 if a negative currency is represented as -12,22 \$
9 if a negative currency is represented as -\$ 12,22
10 if a negative currency is represented as 12,22 \$-

int iCurrency

0 if a currency is represented as \$12,22
1 if a currency is represented as 12,22\$
2 if a currency is represented as \$ 12,22
3 if a currency is represented as 12,22 \$

int iCurrDigits

Number of digits in the currency i.e. 2 for US money 0 for Italian money.

char SZCurrency[5]

Name of the currency.

int iLZero

0 if numbers are represented as ,787 1 if they are represented as 0,787

int iDigits

Number of digits for a number; this value it is NOT used in the number control.

int iMeasure

Unit of measure. Not used in the Library,

char iThousand[2]

Thousand separator i.e. ',' for 12.234,67876

char iDecimal[2]

Decimal Separator i.e. '.' for 12.234,67876

int language

Installation language:

0X0500 for Italian

0X0400 for English

0X0100 for Danish

0X0200 for Finnish

0X0300 for French

0X0600 for Icelandic

0X0700 for Norwegian

0X0800 for Dutch

0X0900 for Portuguese

0X0A00 for Spanish

0X0B00 for Swedish

0X0C00 for German

See Also [GetIntStructure](#) [Set Timezone and Daylight values](#)

intData * FAR PASCAL _export GetIntStructure()

Return an intData structure that must NOT be destroyed.

```
extern "C" void SetEnvTime(LPCSTR env)  
LPSCTR env // New environment variable
```

Set the library timezone, daylight ecc.. to the new values defined by the env environment variable.

For more information consult C documentation.

SET_SHORT_DATE_KEY message

wParam = new Key;

lParam=0L;

Date Control Message

The SET_SHORT_DATE_KEY is send to a date window to modify the key used to insert a short date

Returns 0

NOTE: the key is by default **F8**

NOTE: the new setting is NOT global

SET_CALENDAR_KEY message

wParam = new Key;
lParam=0L;

Date Control Message

The SET_CALENDAR_KEY is send to a date window to modify the key used to display the help calendar.

Returns 0

NOTE: the key is by default **F2**

NOTE: the new setting is NOT global

DS_AUTODATE Style

Date Control style; automatically insert current date in the control.

TS_SECONDS Style

Time Control style: the time control will also display seconds.

TS_AUTOTIME Style

Time Control style: automatically insert current time in the control.

TS_CLOCK

Time Control style: the time behave like a clock.

```
void FAR PASCAL SetSign(hwnd);  
HWND hwnd; /* handle of window */
```

The **Negate** function, if the window is a Currency or Number control put the number to negative.

Returns: null

Example:

The date window contains 123,55

Negate(hwnd);

The window contains -123,55

```
BOOL FAR PASCAL  GetDateTm(hwnd,tm _far *tmIn)
HWND hwnd; /* handle of window */
tm _far *tmIn; /* Address of pointer for time in tm format */
```

The **GetDateTm** function, if the window is a Date control, returns in tmIn the date contained in the window in a tm structure.

Returns: if the date is valid TRUE else FALSE.

Example:

The date window contains 12-3-94

```
tm _far inTm;
```

```
GetDateTm(hwnd,&inTm);
```

Fill the inTm structure.

time_t FAR PASCAL GetDateTime_t(hwnd);
HWND hwnd; /* handle of window */

The **GetDateTime_t** function, if the window is a Date control, returns the date contained in the window as a time_t value.

Returns: if the date is NOT valid 0 is returned.

Example:

The date window contains 12-3-94

time_t in = GetDateTime_t(hwnd);

in contains the time_t value for 12-3-94:00:00:00

BOOL FAR PASCAL **GetString**(hwnd,LPSTR dateIn)
HWND hwnd; /* handle of window */
LPSTRtmIn; /* Address of pointer date string */

The **GetString** function, if the window is a Date control, returns in tmIn the date contained in the window as a string.

Returns: if the date is valid TRUE else FALSE.

Example:

The date window contains 12-3-94

LPSTR date = (LPSTR) GlobalAllocPtr(GMEM_ZEROINIT,12);

GetString(hwnd,date);

Returns in date : '12-3-94';

BOOL FAR PASCAL SetDateTm(hwnd,tm _far tmIn)
HWND hwnd; /* handle of window */
tm _far tmIn; /* Address of pointer to tm structure */

The **SetDateTm** function, if the window is a Date control, set in hwnd the date contained in the In structure.

Returns: if the date is valid TRUE else FALSE.

```
BOOL FAR PASCAL SetDateTime_t(hwnd,time_t In)
HWND hwnd; /* handle of window */
time_t In; /* time_t to which the control must be set */
```

The **SetDateTime_t** function, if the window is a Date control, set in hwnd the date contained in the In value.

Returns: if the date is valid TRUE else FALSE.

BOOL FAR PASCAL SetDateString(hwnd,LPSTR dateIn)
HWND hwnd; /* handle of window */
LPSTR dateIn; /* date to which the control must be set */

The **SetDateString** function, if the window is a Date control, set in hwnd the date contained in the dateIn value.

Returns: if the date is valid TRUE else FALSE.

BOOL FAR PASCAL GetLongDate(tm _far t,LPSTR buff)
tm _far t /* time to be converted */
LPSTR buff /* Address of pointer date string */

The **GetLongDate** function converts a tm structure to a long date string.

Returns: if the tm structure is valid TRUE else FALSE.

LPSTR FAR PASCAL GetDayName(time_t in, LPSTR buff)
tm _far t /* time to be converted */
LPSTR buff /* Address of pointer to Day Name string */

The **GetDayName** function retrieve the Day Name for a time_t value.

Returns: if the time_t is valid TRUE else FALSE.

LPSTR FAR PASCAL GetMonthName(time_t in, LPSTR buff)
tm _far t /* time to be converted */
LPSTR buff /* Address of pointer to Day Name string */

The **GetMonthName** function retrieve the Month Name for a time_t value.

Returns: if the time_t is valid TRUE else FALSE.

```
int FAR PASCAL   GetHours(HWND hwnd);  
HWND hwnd; /* handle of window */
```

The **GetHours** function, if the window is a Time control, returns the given window's Hours value.

```
int FAR PASCAL   GetMinutes( HWND hwnd);  
HWND hwnd; /* handle of window */
```

The **GetMinutes** function, if the window is a Time control, returns the given window's Minutes value.


```
int FAR PASCAL   GetSeconds( HWND hwnd);  
HWND hwnd; /* handle of window */
```

The **GetSeconds** function, if the window is a Time control, returns the given window's Seconds value.

long FAR PASCAL GetDollar(hwnd)
HWND hwnd; /* handle of window */

The **GetDollar** function, if the window is a Currency control, returns the given window's Dollars amount.

Example:

The price window contains US\$ 66,45
long dollars = GetDollars(hwnd);
dollars is equal to 66

Example:

The credit window contains US\$ -66,45
long dollars = GetDollars(hwnd);
dollars is equal to -66

int FAR PASCAL GetCentimes(hwnd)
HWND hwnd; /* handle of window */

The **GetCentimes** function, if the window is a Currency control, returns the given window's Cents amount.

Example:

The price window contains US\$ 66,45
int cents = GetCentimes(hwnd);
cents is equal to 45

The credit window contains US\$ -66,45
int cents = GetCentimes(hwnd);
cents is equal to -45

long FAR PASCAL GetInteger(hwnd)
HWND hwnd; /* handle of window */

The **GetInteger** function, if the window is a Number control, returns the given window's Integer part value.

Example:

The value window contains 2.673.628,8898
long val = GetInteger(hwnd);
dollars is equal to 2.673.628

The value window contains -2.673.628,8898
long val = GetInteger(hwnd);
dollars is equal to -2.673.628

long FAR PASCAL GetFraction(hwnd)
HWND hwnd; /* handle of window */

The **GetFraction** function, if the window is a Number control, returns the given window's Decimal part value.

Example:

The value window contains 2.673.628,8898
long val = GetDecimal(hwnd);
dollars is equal to 8898

The value window contains -2.673.628,8898
long val = GetDecimal(hwnd);
dollars is equal to -8898

```
void FAR PASCAL_export SetDollar(HWND hwnd,long value)
HWND hwnd; /* Handle of the window */
long value; /* Value to set */
```

Set the Dollar value of an currency control to value

```
BOOL FAR PASCAL_export SetHours(HWND hwnd,int value)
HWND hwnd; /* Handle of the window */
long value; /* Value to set */
```

Set the Hours of a time control to value

Return: FALSE is the hour is invalid (>24);

```
BOOL FAR PASCAL_export SetMinutes(HWND,int)
HWND hwnd; /* Handle of the window */
long value; /* Value to set */
```

Set the Minutes of a time control to value

Return: FALSE is the minute is invalid (>60);


```
BOOL FAR PASCAL_export SetSeconds(HWND hwnd ,int value);  
HWND hwnd; /* Handle of the window */  
long value; /* Value to set */
```

Set the Seconds of a time control to value

Return: FALSE is the second is invalid (>60);

```
void FAR PASCAL_export SetInteger(HWND hwnd,long value)
HWND hwnd; /* Handle of the window */
long value; /* Value to set */
```

Set the Integer part of a number control to value

```
void FAR PASCAL_export SetFraction(HWND hwnd ,long value );  
HWND hwnd; /* Handle of the window */  
long value; /* Value to set */
```

Set the Fractional part of a number control to value

```
BOOL FAR PASCAL_export SetCentimes(HWND hwnd ,int value)
HWND hwnd; /* Handle of the window */
long value; /* Value to set */
```

Set the Centimes value of an currency control to value

The Currency Control

A **Currency** control is a rectangular child window in which the user can enter amounts, using the format defined in the Window initialisation file, from the keyboard. The user selects the control and gives it the input focus by clicking the mouse inside it or pressing the TAB key. The user can enter amounts when the control displays a flashing caret. The mouse can be used to move the cursor and select numbers to be replaced or to position the cursor for inserting characters. The BACKSPACE key can be used to delete characters.

The **Currency** control uses the same styles of the Edit control

The Windows initialisation file defines the following properties for a currency:

Positive Representation.

Negative Representation.

Number of decimal digits.

Currency symbol.

The Number of decimal digits property is ignored by this control.

The control accepts number; the - and + signs.

To insert the value 12345,78 just press 12345 before the decimal comma and then 78 with the cursor positioned after the comma; the control is formatted as US\$ 12.345,789 or according to the representation selected in the window control panel.

To put this number to negative press the '-' key.

To return to positive press the '+' key.

To negate a number from a program use the SetSign function

GetDollar GetCentimes SetSign SetDollar SetCentimes

The Number Control

A **Number** control is a rectangular child window in which the user can enter numbers, using the format defined in the Window initialisation file, from the keyboard. The user selects the control and gives it the input focus by clicking the mouse inside it or pressing the TAB key. The user can enter numbers when the control displays a flashing caret. The mouse can be used to move the cursor and select numbers to be replaced or to position the cursor for inserting characters. The BACKSPACE key can be used to delete characters.

The **Number** control uses the same styles of the Edit control.

The Windows initialisation file defines the following properties for a numeric value:

Thousand separator.

Decimal Separator.

Decimal digits.

Initial Zero.

This control just takes in account the first two properties.

The control accepts number; the decimal separator; the - and + signs.

To insert the number 12345,789 just press in sequence 12345,789 the control is formatted as 12.345,789.

To put this number to negative press the '-' key.

To return to positive press the '+' key.

To negate a number from a program use the SetSign function

See also

GetInteger GetFraction SetSign SetFraction SetInteger

The Date Control

A **Date** control is a rectangular child window in which the user can enter dates, using the format defined in the Window initialisation file, from the keyboard. The user selects the control and gives it the input focus by clicking the mouse inside it or pressing the TAB key. The user can enter dates when the control displays a flashing caret. The mouse can be used to move the cursor and select characters to be replaced or to position the cursor for inserting characters. The BACKSPACE key can be used to delete characters. A key may be used to show a small calendar in which the day may be selected.

The Windows initialisation file defines the following properties for a date:

Short Date: Order

Day Format

Month Format

Year Format

Separator

Long Date: Order

Week Day Format

Month Format

Day Format

Year Format

Separator

The Date control is:

Smart: it try to 'understand' the user inputs; for example if in december the user just insert a 3 the date control guess that the user is trying to insert the 3 of december; if the user insert a W (being the language English) the control default to this week wednesday; if the user insert 3 O (or 3 Ott) the control interpret it as 3 October.

Multilingual: the control does not just 'speak' english if p.e. the language selected in the control panel is Italian the control understand Lunedì' for Monday (and the long date will be Lunedì 13 Novembre 1994).

Complete the F2 key (or a key selected by the programmer) displays a small calendar (in the correct language) where the user may select the date. Two keys (redifinable by the programmer) may insert the current day short and long date in the control.

The Date control uses the same styles of the Edit control; an additional style has been defined DS_AUTODATE to fill with the current date the control.

See also:

SET_SHORT_DATE_KEY SET_CALENDAR_KEY

GetDateTm GetDateTime t GetString

SetDateTm SetDateTime t SetDateString

GetLongDate GetDayName GetMonthName

The Time Control

A **Time** control is a rectangular child window in which the user can enter times, using the format defined in the Window initialisation file, from the keyboard. The user selects the control and gives it the input focus by clicking the mouse inside it or pressing the TAB key. The user can enter times when the control displays a flashing caret. The mouse can be used to move the cursor and select numbers to be replaced or to position the cursor for inserting characters. The BACKSPACE key can be used to delete characters.

The Windows initialisation file defines the following properties for a time:

12 and 24 hours formats

Separator

Initial Zero

The time control accept numbers and separator keys.

The Time control uses the same styles of the Edit control; a additional styles has been defined TS_AUTOTIME to fill with the current time the control; TS_SECONDS to define a control that also accept seconds (by default the control does not accept seconds) and TS_CLOCK to create a clock.

See Also

TS_AUTOTIME TS_SECONDS TS_CLOCK

GetHours GetMinutes GetSeconds

SetHours SetMinutes SetSeconds

Introduction

The Internationalisation Control library provides all the controls you need to exactly reflect the setup defined in the windows Control Panel.

The internationalisation link library provides the following controls:

Date control: a smart date control.

Currency control

Time control

Number control

The controls properties are defined in: The intDate Structure

The library is truly international, if you are using the spanish version of window the library speak spanish or portuguese or italian or .. .

The following languages are supported:

Spanish

French

Italian

Portuguese

German

English

Other languages may easily be added.

An ample set of procedures allow to set and retrieve the values inserted in the control.

To use the library:

```
#include "interLib.h"
```

Then either use LoadLibrary in your WinMain function or load the library at start up.

A full demo program is provided with the library.

The Registered version is provided with full source code and FREE of distribution royalty.

[Register](#) [PRICE](#) [OMBUDSMAN Statement](#) [ShareWare Concept](#)
[Other Products](#) [Support](#) [Why To purchase](#)

ORDER

The Internationalisation Controls Library is sold for 30 US\$ through PsL.
The registered Package INCLUDES all the source code.
The registered Package is FREE of distribution royalty.
To Order:

1. Use a CREDIT CARD to register with PSL (ORDERS ONLY!):
PsL accepts MC/VISA, AMEX, and DISCOVER.
Call 800/2424-PSL or 713/524-6394 or
fax your order to 713/524-6398. PsL also accepts orders
via CompuServe at CIS# 71355,470 or by mail to PsL, P. O.
Box 35705, Houston TX 77235-5705. These numbers are for
ORDERING ONLY!! For information about dealer pricing, site
licenses, shipping of products, returns, latest version
numbers or other technical information contact Maurizio Maccani.
- * PSL charges \$4 for shipping,
PSL also accepts international orders, charging \$15 for
shipping.
- * Any comments, suggestions or information:

E-Mail: maurizio.maccani%h1bbs@ibase.org.br
or
Maurizio Maccani +55.21.256-2464
or
PO.BOX 10893
Copacabana
RIO DE JANEIRO
BRASIL

PRICE

The Internationalisation Controls Library is sold for 30 US\$ through PsL.

The registered Package **INCLUDES** all the source code.

With the registered package You get a 6 month free upgrade bonus.

When Window 95 is released you will get for **FREE** the new version

ShareWare

Most money back guarantees work like this: You pay for the product and then have some period of time to try it out and see whether or not you like it. If you don't like it or find that it doesn't do what you need, you return it (undamaged) and at some point - which might take months - you get your money back. Some software companies won't even let you try their product! In order to qualify for a refund, the diskette envelope must have an unbroken seal. With these "licensing" agreements, you only qualify for your money back if you haven't tried the product. How absurd!

Shareware is very different. With shareware you get to use it for a limited time, without spending a penny. You are able to use the software on your own system(s), in your own special work environment, with no sales people looking over your shoulder. If you decide not to continue using it, you throw it away and forget all about it. No paperwork, phone calls, or correspondence to waste your valuable time. If you do continue using it, then - and only then - do you pay for it.

Shareware is a distribution method, NOT a type of software. Shareware is produced by accomplished programmers, just like retail software. There is good and bad shareware, just as there is good and bad retail software. The primary difference between shareware and retail software is that with shareware you know if it's good or bad BEFORE you pay for it.

The shareware system and the continued availability of quality shareware products depend on your willingness to register and pay for the shareware you use. It's the registration fees you pay which allow us to support and continue to develop our products.

Please show your support for shareware by registering those programs you actually use and by passing them on to others.

Shareware is kept alive by YOUR support!

Ombudsman Statement

This program is produced by a member of the Association of Shareware Professionals (ASP). ASP wants to make sure that the shareware principle works for you. If you are unable to resolve a shareware-related problem with an ASP member by contacting the member directly, ASP may be able to help. The ASP Ombudsman can help you resolve a dispute or problem with an ASP member, but does not provide technical support for members' products. Please write to the ASP Ombudsman at 545 Grover Rd., Muskegon, MI USA, or send a CompuServe message via CIS MAIL to ASP Ombudsman 72050,1433.

Shows the values in the intData structure

This control has been created using the following call:

```
CreateWindow("Currency", "545,3", WS_BORDER | WS_CHILD | WS_VISIBLE |  
ES_AUTOHSCROLL, 120, 20, 100, 25, hwnd, 1, hInst, NULL);
```

This control has been created using the following call:

```
CreateWindow("Currency","",WS_BORDER | WS_CHILD | WS_VISIBLE |  
ES_AUTOHSCROLL,120,50,100,25,hwnd,1,hInst,NULL);
```


This control has been created using the following call:

```
CreateWindow("Currency", "-545,3", WS_BORDER | WS_CHILD | WS_VISIBLE |  
ES_AUTOHSCROLL, 120, 80, 100, 25, hwnd, 1, hInst, NULL);
```

This control has been created using the following call:

```
CreateWindow("Currency","-",WS_BORDER | WS_CHILD | WS_VISIBLE |  
ES_AUTOHSCROLL ,120,110,100,25,hwnd,1,hInst,NULL);
```

This control has been created using the following call:

```
CreateWindow("Date", "", WS_BORDER | WS_CHILD | WS_VISIBLE  
, 120, 20, 100, 25, hwnd, 1, hInst, NULL);
```

This control has been created using the following call:

```
CreateWindow("Date", "", DS_AUTODATE | WS_BORDER | WS_CHILD | WS_VISIBLE  
, 120, 50, 100, 25, hwnd, 1, hInst, NULL);
```

This control has been created using the following call:

```
CreateWindow("Date", "", WS_BORDER | WS_CHILD | WS_VISIBLE  
, 120, 80, 100, 25, hwnd, 1, hInst, NULL);
```

This control has been created using the following call:

```
CreateWindow("Time","",WS_BORDER | WS_CHILD | WS_VISIBLE  
,120,20,130,25,hwnd,1,hInst,NULL);
```

This control has been created using the following call:

```
CreateWindow("Time","",TS_SECONDS | WS_BORDER | WS_CHILD | WS_VISIBLE  
,120,50,130,25,hwnd,1,hInst,NULL);
```

This control has been created using the following call:

```
CreateWindow("Time","", TS_AUTOTIME | TS_SECONDS | WS_BORDER | WS_CHILD |  
WS_VISIBLE,120,80,130,25,hwnd,1,hInst,NULL);
```


This control has been created using the following call:

```
CreateWindow("Time","", TS_CLOCK | TS_SECONDS | WS_CHILD | WS_VISIBLE |  
ES_READONLY,120,110,130,25,hwnd,1,hInst,NULL);
```

This control has been created using the following call:

```
CreateWindow("Time","", TS_CLOCK | WS_CHILD | WS_VISIBLE |  
ES_READONLY,120,140,130,25,hwnd,1,hInst,NULL);
```

This control has been created using the following call:

```
CreateWindow("Number", "333,88", WS_BORDER | WS_CHILD | WS_VISIBLE  
, 120, 20, 100, 25, hwnd, 1, hInst, NULL);
```

This control has been created using the following call:

```
CreateWindow("Number","",WS_BORDER | WS_CHILD | WS_VISIBLE  
,120,50,100,25,hwnd,1,hInst,NULL);
```

This control has been created using the following call:

```
CreateWindow("Number", "-333,88", WS_BORDER | WS_CHILD | WS_VISIBLE  
, 120, 80, 100, 25, hwnd, 1, hInst, NULL);
```

Exit Demo

Display a dialog with some controls

Change the default language to ...

Call help file for ...

Read the information contained in the currently displayed dialogues.

Plays a control demo

This control has been created using the following dialogue statement:

```
CONTROL      "",1,"Currency",WS_BORDER | WS_TABSTOP,100,10,80,12
```

This control has been created using the following dialogue statement:

```
CONTROL      "",2,"Number",WS_BORDER | WS_TABSTOP,100,25,80,12
```

This control has been created using the following dialogue statement:

```
CONTROL      "",3,"Date",WS_BORDER | WS_TABSTOP,100,40,50,12
```

This control has been created using the following dialogue statement:

```
CONTROL      "",4,"Time",WS_BORDER | WS_TABSTOP,100,55,50,12
```

This control has been created using the following dialogue statement:

```
CONTROL      "",5,"Time",WS_BORDER | WS_TABSTOP |  
TS_SECONDS,100,70,50,12
```


This control has been created using the following dialogue statement:

```
CONTROL          "555,66",1,"Currency",WS_BORDER | WS_TABSTOP,100,10,80,12
```

This control has been created using the following dialogue statement:

```
CONTROL      "555,66",2,"Number",WS_BORDER | WS_TABSTOP,100,25,80,12
```

This control has been created using the following dialogue statement:

```
CONTROL      "" ,3,"Date",WS_BORDER | DS_AUTODATE |  
WS_TABSTOP,100,40,50,12
```

This control has been created using the following dialogue statement:

```
CONTROL      "12 3 98",4,"Date",WS_BORDER | WS_TABSTOP,100,55,50,12
```

This control has been created using the following dialogue statement:

```
CONTROL      "",5,"Time",WS_BORDER | WS_TABSTOP,100,70,50,12
```

This control has been created using the following dialogue statement

```
CONTROL          "",6,"Time",WS_BORDER | WS_TABSTOP |  
TS_AUTOTIME,100,85,50,12
```

This control has been created using the following dialogue statement:

```
CONTROL          "12 5 7",7,"Time",WS_BORDER | WS_TABSTOP |  
TS_SECONDS,100,100,50,12
```

Support

Registered users are entitled to 6 months free support by E-mail or telephone.
For support please contact:

maurizio.maccani%hlbbs@ibase.org.br

or

+55 (21) 256-2464

Other Shareware Products by Maurizio Maccani

Internationalization Library, V.1.0 <ASP> -

Time, Date, Number and Currency formatted controls library for Windows.
INTLB010.ZIP

Time, Date, Number and Currency formatted controls library for Windows;
the formatting rules are read from the control panel setup.

The date control is 'multilingual' (answers in various language) and
'smart' (accepts partial entries as W: this week Wednesday or F 4: the forth
of February); the date control includes a monthly calendar control.

The Time control may be used as a clock with second or minute precision.

The Currency control formats to positive, negative amounts.

Full sources included - Royalty free

US\$ 30

Internationalization VBX, V.1.0 <ASP> -

The Internationalization control as a Visual Basic Library.

Forecasted release date March 1995

INTVB010.ZIP

Telephone Control, V.1.0 <ASP> -

A smart formatted telephone number control. Automatically formats/dial
telephone numbers.

TELLB010.ZIP

Contains an user updatable Data Base with the main country / area telephone
codes;

It allows telephone number dial with a simple keystroke; the dialling is 'smart'
i.e. area and international prefix may automatically be added when the
product is used from a different town or country (as happen with portable).

Full sources included - Royalty free

US\$ 50

Telephone VBX, V.1.0 <ASP> -

The telephone library as a Visual Basic Library.

Forecasted release date March 1995

Window Analyser V1.0 <ASP> -

A tool for Windows programmers. With WA it is possible to analyse all
the principal characteristics of a Windows control.

WA010.ZIP

Window Analyser reports:

- Control Class properties and styles
- Control Characteristics
- Control Properties
- Control Extra Memory (Dump included)
- Control Styles
- Control Parent and Childs..

US\$ 30

Software Training Package V1.0 <ASP> -

A tool to quickly build Training and Demo packages for software products.
STP010.ZIP

It allows To quickly build training and demo scripts for Your software package.
No programming skills are needed: start the product; register your demo/training
play it again to tune it; release it. Training may be called from inside your
product or from an Help File.
Very small add just a 10K to your application

Royalty free

US\$ 70

SuperSpy V1.0 <ASP> -

An advanced spy utility.
SSPY010.ZIP

- Allows to include your Own mnemonics i.e. if you defined a WM_DO_MY_STUFF
message as WM_USER + 124 Superspy shows a WM_DO_MY_STUFF message not a
WM_USER+124

- Interpret wParam and lParam; if it is possible extracts the structure
pointed by lParam and get the values or display the values in a meaningful way.
p.e. the WM_ACTIVATE parameters are shown as :

wParam = WA_INACTIVE
window was inactive
window de-activated 12345

US\$ 30

Easy Context Sensitive Help V1.0 <ASP> -

Build your context sensitive help.

ECS010.zip

With just 4 call you enable your application to provide context sensitive
help.

- SetMenuHelp(menuHandle,helpTopic)
- SetWindowHelp(windowHandle,helpTopic)
- SetDefaultHelp(helpTopic)
- SetHelpFile(helpFileName)

This library will automatically manage all the details of building a Context Sensitive Help from intercepting F1 to searching the help for the parent window if no help for the current window / menu item is encountered.

FREEWARE - BANNERWARE

Royalty free

Maurizio Maccani Productivity Pack V.1.0 <ASP>

MMPP010.ZIP

Contains all the above products at a discount price: US\$ 149

Special Offer if ordered before July 95 US\$ 99 .

The Productivity pack will contain, when released, the two VBX controls.

All these package grant free upgrade to Window 95 version.

All these packages may be copied from:

the major BBS and FTP sites.

Compuserve

CICA, SimTel, Garbo and their mirror sites.

All these products may be ordered to PsL

PsL accepts MC/VISA, AMEX, and DISCOVER.

Call 800/2424-PSL or 713/524-6394 or

fax your order to 713/524-6398. PsL also accepts orders

via CompuServe at CIS# 71355,470 or by mail to PsL, P. O.

Box 35705, Houston TX 77235-5705. These numbers are for

ORDERING ONLY!! For information about dealer pricing, site

licenses, shipping of products, returns, latest version

numbers or other technical information contact Maurizio Maccani.

Why to Purchase

- A) You do the right thing
- B) You get support
- C) You are authorised to distribute the DLL royalty free
- D) The Monthly calendar screen title bar unregistered message disappears
- E) You get the FULL sources
- F) You get a 6 months free upgrades
- G) When Window 95 is released you will get a free copy

For Other interesting products see the [Other Products](#) topic

