

## **Index de l'aide YRTRACE 3.1**

Configuration et liste des fichiers

Que fait YRTRACE ?

Quoi de nouveau dans la version 3.1? et limites du shareware

Options

Contenu de la liste de messages

Les erreurs détectées

Les premiers pas...

Classification des messages

Liaison avec vos logiciels

Fonctionnement interne

Les macros et le YRBasic

Je cherche un job !

## Liaison avec vos logiciels

Vous pouvez créer un lien entre YRTRACE et vos programmes. Ces liens peuvent vous permettre de suivre vos programmes au fur et à mesure de leurs déroulements. Couplé à la capture de messages et d'erreurs, cette possibilité devient vite utile.

Ce lien s'obtient via la DLL YRHOOK31.DLL et ceci de façon classique, soit via un fichier d'import, soit par index ou par nom (reportez-vous au manuel du langage que vous utilisez). Le fichier YRHOOK31.LIB ainsi qu'un exemple EX\_BCPP (voir plus particulièrement les fichiers EX\_BCPP.DEF et YR\_DLL.H.

### Les index et les fonctions sont les suivants:

#### **BOOL FAR PASCAL IsYRTRACE()**

Retourne true si YRTRACE est chargé.

Index:6

#### **BOOL FAR PASCAL IsRegister()**

Retourne TRUE si la version de YRTRACE est enregistrée.

Index:7

#### **void FAR PASCAL YRAddline(LPSTR txt)**

Ajoute une ligne de commentaire contenue dans txt dans la liste de YRTRACE et (#) sur le terminal (Minitel) si celui si est connecté.

Index:8

#### **void FAR PASCAL ClearList()**

Efface le contenu de la liste de YRTRACE

Index:9

#### **void FAR PASCAL StartCapture()**

(#) Départ de la capture de messages de YRTRACE

Index:10

#### **void FAR PASCAL StopCapture()**

(#) Arrêt de la capture de messages de YRTRACE

Index:11

#### **void FAR PASCAL ShowYRTrace()**

(#) Provoque la mise au premier plan de la fenêtre YRTRACE

Index:12

#### **void FAR PASCAL HideYRTrace()**

(#) Cache la fenêtre YRTRACE

Index:13

#### **void FAR PASCAL RunYRBASIC(LPSTR prog)**

(#) Lance le programme YRBASIC dont le chemin\nom est contenu dans prog

Index:14

**Notes:** "(#)" désigne des options uniquement utilisables après enregistrement.  
la DLL est écrite en C++

## **Enregistrement gratuit de YRTRACE**

L'enregistrement est gratuit. Son objet est uniquement de savoir si YRTRACE est un outil privé ou s'il intéresse un cercle plus grand de développeurs, au quel cas, il sera amélioré et continuera à être diffusé dans ses nouvelles versions.

De plus, YRTRACE peut faire l'objet d'améliorations à travers des idées, des critiques ou des remarques que vous pouvez me formuler à son encontre.

Aussi, et pour bénéficier des fonctions bridées, vous pouvez l'enregistrer en envoyant une enveloppe timbrée à votre nom (et pourquoi pas une carte postale) à

Didier Heiderich  
2, résidence Beauséjour  
rue Bach  
38120 Le Fontanil

vous recevrez en retour les codes permettant de transformer votre version en version enregistrée et débridée.

Merci.

## **Quoi de nouveau dans la version 3.1? et limites du shareware**

### **1 - Bogues dans YRTRACE:**

Dans la version actuelle, les problèmes rencontrés avec les options "filtre des fenêtres" et "filtre des applications" sont résolus.

**2 - Une option du menu** et un bouton vous permettent d'accéder directement à l'édition d'un programme YRBASIC via le bloc-notes de Windows.

**3 - YRTRACE traite dorénavant les messages multimédia.**

**4 - (#) Vous pouvez supprimer (à vos risques et périls) un module** (DLL, VBX, etc...) de la mémoire via le dialogue de visualisation des modules. Cette option est très utile lors de la conception de DLLs.

**5 - L'innovation la plus importante est la possibilité de relier YRTRACE avec vos applications:**

- vous pouvez ajouter une ligne dans YRTRACE ou sur le Minitel (sur le Minitel à condition d'avoir enregistré le logiciel), ce qui vous permet de suivre à la trace vos logiciels au moment du déroulement du programme. Couplé à la capture de messages et d'erreurs cette nouvelle possibilité devient très vite utile dans le débogage d'applications.

- (#) pour améliorer ce principe, des instructions permettent de déclencher et d'arrêter la capture de messages, de visualiser ou cacher YRTRACE ou encore le lancer une macro YRBASIC. La liaison se fait classiquement via une DLL.

**Note:** (#) ces options sont disponibles uniquement après enregistrement du logiciel (ce qui est gratuit). Il n'y a aucune limite dans les options déjà disponibles dans la version précédente.

### **Pourquoi des limites dans cette version ?**

Malgré la gratuité de l'enregistrement de la version précédente, un seul utilisateur m'a fourni des encouragements. Aussi, cette nouvelle version existe uniquement car elle satisfait mes propres besoins...Ce serait dommage de ne pas continuer à améliorer YRTRACE si vous l'utilisez!

## Capture des messages

**Départ** provoque la capture (selon les options), de même que l'instruction YRBASIC "YRRUN ON"

**Dans ce cas, il arrive que YRTRACE modifie le curseur de l'application pour signaler qu'il capture des messages en même temps que l'application fonctionne.**

**Stop** arrête la capture, de même que l'instruction YRBASIC "YRRUN OFF"

Voir :

[Contenu de la liste de messages](#)

[Options](#)

[Classification des messages](#)

[Les erreurs détectées](#)

## **Sélection des applications**

A L'EXCEPTION DES ERREURS, YRTRACE vous permet de choisir les applications concernées par la capture des messages, les autres seront ignorées.

Voir :

[Contenu de la liste de messages](#)

[Options](#)

[Classification des messages](#)

## **Sélection des fenêtres**

YRTRACE vous permet de sélectionner :

- toutes les fenêtres,
- une fenêtre :

dans ce cas, YRTRACE disparaît, vous sélectionnez la fenêtre en cliquant avec le bouton gauche.

Seule(s) la (les) fenêtre(s) sélectionnée(s) fait (font) l'objet de la capture des messages.

Contenu de la liste de messages

Options

Classification des messages

## **Sélection des messages**

YRTRACE vous permet de sélectionner les messages capturés et ceci par type de messages.

## **Définir visualisation**

**Toujours visible** : YRTRACE reste au premier plan de l'écran. Cette option peut être modifiée par l'instruction YRBASIC : SHOWWINDOW

**La liste suit les messages** : le dernier message windows est visible dans la liste. Le contenu de la liste peut être modifié par les instructions YRBASIC : YRADDLINE et YRCLEARLIST

**Montrer dans la liste** : définit ce qui sera visible dans la liste.

**Erreurs** : active la détection des erreurs.

## **Associer Macro**

Vous permet d'associer un programme YRBASIC à un message particulier.

Si la détection de messages est activée, selon les conditions de capture, la détection de ce message chargera et exécutera la macro YRBASIC.

La macro s'exécutera en parallèle de YRTRACE (si vous avez décidé de ce fonctionnement en utilisant l'instruction basic : MULTITASK ON).

Dans le dialogue, les flèches renvoient au prochain message ayant une macro associée (selon le sens).

Voir :

[Fonctionnement interne](#)

[Macros en YRBASIC](#)

## Modules

Lors de la mise au point d'applications, il est parfois utile de connaître les modules en mémoire, cette option ouvre un dialogue affichant les modules en cours. Vous pouvez sélectionner le type de module visible en utilisant le bouton "option" du dialogue. La version 3.1 enregistrée permet également de **supprimer un module (DLL, VBX, DRV, ...)** de la mémoire. Cette option peut s'avérer très utile lors de la conception de DLLs.

Même si YRTRACE le permet, il est fortement déconseillé d'éliminer de la mémoire des modules sans savoir précisément ce que vous êtes en train de faire...

## **YRTRACE et les communications sur les ports séries**

### **Généralités**

YRTRACE vous permet d'envoyer ses données vers le port série, ainsi vous pouvez suivre la capture des messages et des erreurs sans encombrer l'écran de votre PC.

Pour cela, il vous faut un Minitel (ou un autre terminal) au standard téléinformatique 80 colonnes et jeux américain.

Pour obtenir le standard téléinformatique sur le Minitel (de 1B à 2) : Fnct T + A

Si vous possédez un Minitel 1B, vous pourrez établir la communication en 1200, paire, 7, 1. Si vous avez un Minitel 2, ce qui est recommandé, vous pouvez choisir la vitesse de communication, dans ce cas sélectionnez 9600 dans le dialogue et modifiez la vitesse de la prise sur votre Minitel en utilisant : Fnct P + 9.

En cas d'échec, vérifiez sur la documentation du Minitel les modes supportés et les moyens d'y accéder. Contrôlez également le n° du port série et si aucun autre logiciel est en train de l'utiliser.

YRTRACE ne supporte qu'une seule communication.

Les communications sont également accessibles via le YRBASIC et directement en relation avec la communication du programme : donc, si le YRBASIC provoque la communication et que vous êtes en détection de messages, les messages s'afficheront sur votre terminal.

### **Options**

Les options sont classiques. La sélection d'un terminal provoque simplement des messages circonstanciés.

## Les premiers pas...

Pour avoir une première approche de YRTRACE :

Lancez YRTRACE.

- Dans le menu "Options", sélectionnez : "Définir visualisation". Cochez la case "Toujours visible", cliquez sur "OK".
- Dans le menu "Options", choisissez "Messages", cochez le dialogue "Menus".
- Dans le menu "Capture", choisissez "Départ".

Ensuite, allez dans le menu d'une application autre que YRTRACE, vous verrez les messages apparaître dans la liste selon les attributs définis dans "Visualisation".

## Les erreurs détectées

**La détection des erreurs est effectuée via les outils fournis en standard par Windows, cette détection est par conséquent sûre, y compris lorsque l'erreur concerne un pointeur.**

### Erreurs dans les appels de fonctions Windows

'GlobalAlloc' A ECHOUÉ  
'GlobalReAlloc' A ECHOUÉ  
'GlobalLock' A ECHOUÉ  
'LocalAlloc' A ECHOUÉ  
'LocalReAlloc' A ECHOUÉ  
'LocalLock' A ECHOUÉ  
'AllocResource' A ECHOUÉ"  
'LockResource' A ECHOUÉ"  
'LoadModule' A ECHOUÉ"  
'CreateDialog()' A ECHOUÉ: impossible de charger le menu.  
'CreateDialog()' A ECHOUÉ: CREATION DE FENETRE ECHOUÉ  
'RegisterClass' A ECHOUÉ: LA CLASSE EXISTE DÉJÀ  
Le cache des DC est plein: trop de 'GetDC() !  
'CreateWindow' A ECHOUÉ: CLASSE DE FENETRE INTROUVABLE  
DESALLOCATION DE FENETRE EXTRA / MOT DE CLASSE INVALIDE  
'LoadString()' A ECHOUÉ  
LoadMenu()' A ECHOUÉ  
Superposition des appels 'BeginPaint()  
INDEX DE FENETRE INVALIDE  
'CreateMenu' A ECHOUÉ  
CreateDC A ECHOUÉ  
'CreateMetafile': CREATION DU METAFIÈRE ECHOUÉ  
"BITMAP DÉJÀ SÉLECTIONNÉ  
L'OBJET SÉLECTIONNÉ EST DÉTRUIT.

### Erreurs sur des objets mémoires

YRTRACE envoi un message INVALIDE appliqué à :

"valeur", "index", "drapeau", "selecteur", "pointeur", "pointeur de fonction", "pointeur de chaîne", "HINSTANCE", "HMODULE", "global handle", "local handle", "atom", "HWND", "HMENU", "HCURSOR", "HICON", "HGDIOBJ", "HDC", "HPEN", "HFONT", "HBRUSH", "HBITMAP", "HRGN", "HPALETTE", "HANDLE", "HFILE", "HMETAFIÈRE", "CID", "HDRV", "HDWP".

## Fonctionnement interne

Traitements en parallèle	Conditions	Résultat
<p><u>Arrivé d'un message Windows</u></p> <p>CAPTURE DU MESSAGE</p>	<p><b>si</b></p> <ul style="list-style-type: none"> <li>- pas de réentrance</li> <li>- fenêtre sélectionnée</li> <li>- application sélectionnée</li> <li>- groupe du message sélectionné</li> </ul> <p><b>si</b></p> <ul style="list-style-type: none"> <li>- pas de programme Basic associé à un message est en cours</li> <li>- programme Basic associé au message</li> </ul>	<p>Affichage dans la liste écran et communication (si communication établie)</p> <p>Lancement du programme BASIC associé</p>
MACRO BASIC EN FONCTIONNEMENT	<p><b>si</b></p> <ul style="list-style-type: none"> <li>- fichier Basic existe</li> <li>- pas d'erreur</li> </ul>	Selon le programme
CAPTURE DES ERREURS	<p><b>si</b></p> <ul style="list-style-type: none"> <li>- pas de réentrance</li> <li>- capture des erreurs sélectionnée</li> </ul>	Affichage dans la liste et port série
MACRO BASIC LANCEE PAR L'UTILISATEUR	<p><b>si</b></p> <ul style="list-style-type: none"> <li>- pas de réentrance</li> <li>- pas d'erreur</li> </ul>	Selon le programme

Le schéma ci-dessus montre les possibilités et les limites de YRTRACE :

- Plusieurs programmes en BASIC peuvent fonctionner en même temps ! En effet, l'interpréteur est un objet, ce qui explique cette possibilité. Cependant, pour éviter la multiplication des programmes en mémoire, et de là, des problèmes de toute sorte, YRTRACE limite le nombre de programmes simultanés en interdisant les réentrances.

De fait seulement 3 programmes peuvent fonctionner en même temps: le programme en tâche de fond, un programme déclenché par un message, et le testeur de programme situé dans l'option "associer une macro".

La réentrance concerne uniquement chacun des éléments séparés horizontalement ci-dessus, indépendamment, ils sont pseudo-multitâches.

- L'absence de réentrance peut vous empêcher de détecter certains messages ou certaines erreurs, il suffit pour cela que l'événement se produise pendant le traitement d'un événement précédent.

## Configuration

Il vous faut

- au minimum un 386 avec 2 Mo de mémoire et un écran couleur à partir d'EGA.
- Windows 3.1

## Liste des fichiers

YRTRACE est composé de 13 fichiers inséparables :

Fichiers YRTRACE :

YR31.EXE  
YRHOOK31.DLL  
YR31.HLP  
YR31.WRI

Bibliothèques BORLAND:

BC402RTL .DLL  
BIDS402.DLL  
BWCC.DLL  
OWL202.DLL

Bibliothèque Microsoft :

CTL3DV2.\_\_\_\_

Fichier à utiliser dans vos programmes en C et C++ :

yrhook31.lib

Programmes YRBASIC :

MNTCTRL.BYR  
plus les exemples : \*.BYR

Démonstration de liaison Logiciel/YRTRACE :

ex\_bcpp.exe  
ex\_link.byr  
plus le code source de la démonstration..

(Vous devez posséder TOOLHELP.DLL normalement livré en standard avec Windows (\*) )

## Que fait YRTRACE ?

**YRTRACE est un espion** qui traque les messages de Windows () et les affiche dans une boîte de liste ou sur le Minitel relié à votre PC,

**YRTRACE est un débogueur** qui détecte de nombreuses erreurs se produisant dans les applications de Windows,

**YRTRACE est interactif**; via une DLL, vos logiciels peuvent l'utiliser, par exemple, pour suivre le déroulement de vos applications sur YRTRACE en avant plan **et sur le Minitel** ou encore en déclenchant un programme YRBASIC, couplé à la capture de messages et d'erreurs cette possibilité devient très vite utile dans le débogage d'applications,

**YRTRACE est un outil** qui permet de connaître les ressources disponibles, les modules en mémoire et de **supprimer des modules** de la mémoire (DLL, VBX, DRV,...), 

**YRTRACE est puissant**; il intègre le **YRBASIC** langage proche du BASIC mais spécialisé pour Windows et YRTRACE. Un programme YRBASIC peut être lancé via le menu, par l'arrivée d'un message, ou encore depuis vos logiciels. Les programmes YRBASIC peuvent être pseudo-multitâches et utiliser votre Minitel. Une application est livrée à cet effet : MNTCTRL.BYR : essayez-la !

**Note:** Le fonctionnement (capture des messages et des erreurs) est basé sur des fonctions internes de Windows prévues à cet effet.

## Contenu de la liste de messages

Dans la liste de messages, on retrouve dans l'ordre suivant :

- Le nom de l'application
- Le nom ou le numéro du message
- Le numéro de la fenêtre concernée par le message
- Le nom de la classe de la fenêtre
- Le nom de la fenêtre
- Les paramètres du message :
  - \* WParam
  - \* LParam Lo
  - \* LParam Hi
  - \* LParam
- Eventuellement, le texte associé à LParam
- Les ressources disponibles

Le nombre maximum de messages dans la liste est de 300. Une fois ce nombre de messages atteint, la liste est amputée de son premier message.

## **Classes de fenêtres**

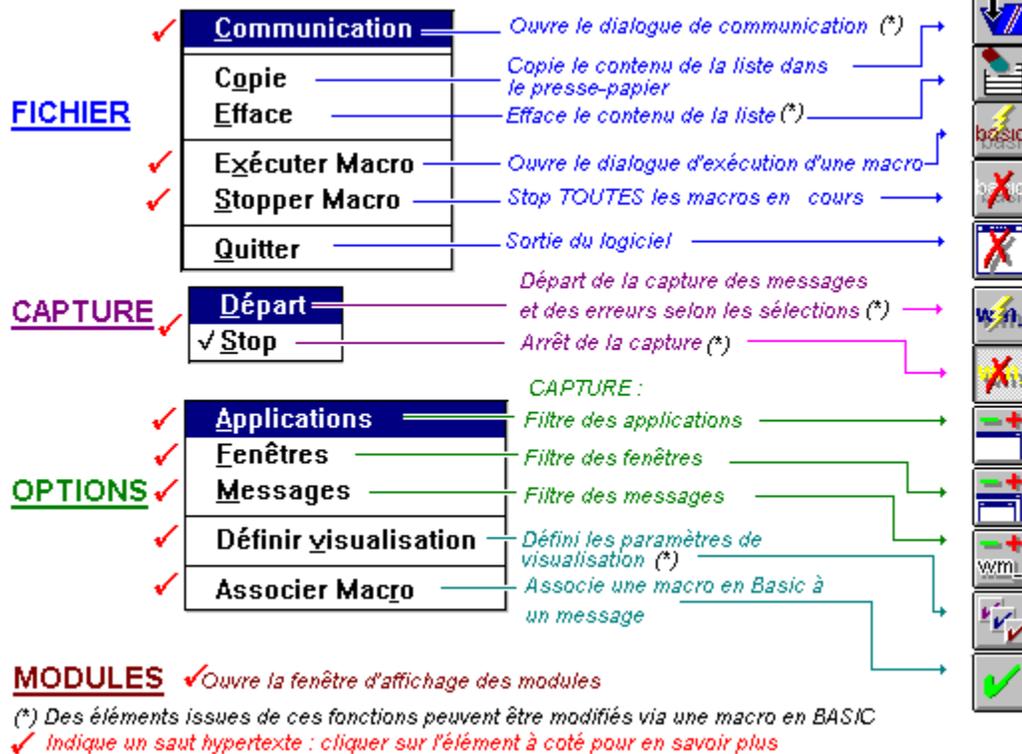
Chaque fenêtre fait partie d'une classe. YRTRACE 3.1 peut afficher le nom de la classe de fenêtre (parfois le nom est juste un #n°).

Certains messages de nom différents ont le même numéro : YRTRACE se sert du nom de la classe pour afficher le nom du message qui lui paraît correcte.

Nom de classes reconnues et permettant la différenciation:

Button, BorRadio, ListBox, ComboBox, ComboLBox, #32768 (menus).

## Options de YR TRACE



## Editer macro

lance le bloc-notes pour éditer une macro



## Classification des messages

Les messages sont répertoriés dans YR TRACE selon 18 classes :

- Edition : messages concernant l'édition de texte.
- Boîtes de liste : messages utilisés par les boîtes de liste.
- Icône curseur : messages concernant les icônes et les curseurs
- Bouton et Case à cocher : messages boutons et des boîtes à cocher
- Listes déroulantes : messages utilisés par les listes déroulantes
- DDE : messages client / serveur
- Clavier : messages utilisés pour la gestion du clavier
- Souris : messages utilisés pour la souris
- Dessine : messages concernant le dessin des fenêtres
- MDI : messages utilisés pour la gestion des fenêtres MDI
- Menu : messages utilisés par les menus
- Système : messages des menus et paramètres systèmes
- Presse papier et imprimante : messages du presse-papier et du tampon d'impression
- Application Fenêtres : messages de gestion des applications et des fenêtres (sauf MDI)
- WM\_COMMAND : message WM\_COMMAND
- Autres : messages non classés
- Utilisateurs : messages définis par les utilisateurs (entre WM\_USER et \$7FFF)
- Multimédia : messages multimédias
- Non répertoriés : messages non répertoriés (YRTRACE fourni le n° du message).

## **Messages traqués par l'option Multimédia**

MM\_JOY1MOVE

MM\_JOY1MOVE

MM\_JOY2MOVE

MM\_JOY2MOVE

MM\_JOY1ZMOVE

MM\_JOY2ZMOVE

MM\_JOY1BUTTONDOWN

MM\_JOY2BUTTONDOWN;

MM\_JOY1BUTTONUP

MM\_JOY2BUTTONUP

MM\_JOY2BUTTONUP

MM\_MCINOTIFY

MM\_WOM\_OPEN

MM\_WOM\_CLOSE

MM\_WOM\_DONE

MM\_WIM\_OPEN

MM\_WIM\_CLOSE

MM\_WIM\_DATA

MM\_MIM\_OPEN

MM\_MIM\_CLOSE

MM\_MIM\_DATA

MM\_MIM\_LONGDATA

MM\_MIM\_ERROR

MM\_MIM\_LONGERROR

MM\_MOM\_OPEN

MM\_MOM\_CLOSE

MM\_MOM\_DONE

## **Messages traqués par l'option AUTRES**

WM\_COMMNOTIFY

WM\_DROPFILES

WM\_NCHITTEST

WM\_POWER

WM\_QUEUESYNC

WM\_COMPACTING

WM\_ENTERIDLE

## **Messages traqués par l'option TIMER**

WM\_TIMECHANGE

WM\_TIMER (Attention : WM\_TIMER est un message non prioritaire)

## **Messages traqués par l'option APPLICATION & FENETRES**

WM\_ACTIVATE

WM\_ACTIVATEAPP

WM\_CLOSE

WM\_CANCELMODE

WM\_CHILDACTIVATE

WM\_CREATE

WM\_ENDSESSION

WM\_INITDIALOG

WM\_DESTROY

WM\_HSCROLL

WM\_DEVMODECHANGE

WM\_KILLFOCUS

WM\_VSCROLL

WM\_GETMINMAXINFO

WM\_GETDLGCODE

WM\_QUERYDRAGICON

WM\_QUERYENDSESSION

WM\_QUERYNEWPALETTE

WM\_QUERYOPEN

WM\_SETFOCUS

WM\_PARENTNOTIFY

WM\_WININICHANGE

WM\_QUIT

WM\_MOVE

## **Messages traqués par l'option SYSTEME**

WM\_SYSCOLORCHANGE

WM\_SYSCOMMAND

WM\_SYSDEADCHAR

WM\_SYSTEMERROR

DM\_GETDEFID

DM\_SETDEFID

WM\_SYSKEYDOWN

WM\_SYSKEYUP

WM\_SYSCHAR

## **Messages traqués par l'option MDI**

WM\_MDIACTIVATE

WM\_MDICASCADE

WM\_MDICREATE

WM\_MDIDESTROY

WM\_MDIGETACTIVE

WM\_MDIICONARRANGE

WM\_MDIMAXIMIZE

WM\_MDINEXT

WM\_MDIRESTORE

WM\_MDISETMENU

WM\_MDITILE

## **Messages traqués par l'option PAINT**

WM\_NEXTDLGCTL

WM\_NCPAINT

WM\_PAINT

WM\_PALETTECHANGED

WM\_PALETTEISCHANGING

WM\_SIZE

WM\_WINDOWPOSCHANGED

WM\_WINDOWPOSCHANGING

WM\_ERASEBKGD

WM\_DRAWITEM

WM\_CTLCOLOR

WM\_FONTCHANGE

WM\_MEASUREITEM

WM\_NCACTIVATE

WM\_NCCALCSIZE

WM\_NCCREATE

WM\_NCDESTROY

WM\_GETFONT

WM\_SETFONT

WM\_SETREDRAW

WM\_SHOWWINDOW

## **Messages traqués par l'option SOURIS**

WM\_LBUTTONDOWNBLCLK

WM\_LBUTTONDOWN

WM\_LBUTTONUP

WM\_MBUTTONDOWNBLCLK

WM\_MBUTTONDOWN

WM\_MBUTTONUP

WM\_NCLBUTTONDOWNBLCLK

WM\_NCLBUTTONDOWN

WM\_NCLBUTTONUP

WM\_NCMBUTTONDBLCLK

WM\_NCMBUTTONDOWN

WM\_NCMBUTTONUP

WM\_NCMOUSEMOVE

WM\_NCRBUTTONDOWNBLCLK

WM\_NCRBUTTONDOWN

WM\_NCRBUTTONUP

WM\_RBUTTONDOWNBLCLK

WM\_RBUTTONDOWN

WM\_RBUTTONUP

WM\_MOUSEACTIVATE

WM\_MOUSEMOVE

## **Messages traqués par l'option CLAVIER**

WM\_KEYDOWN

WM\_KEYUP

WM\_DEADCHAR

WM\_VKEYTOITEM

WM\_CHAR

WM\_CHARTOITEM

## **Messages traqués par l'option DDE**

WM\_DDE\_ACK

WM\_DDE\_ADVISE

WM\_DDE\_DATA

WM\_DDE\_EXECUTE

WM\_DDE\_INITIATE

WM\_DDE\_POKE

WM\_DDE\_REQUEST

WM\_DDE\_TERMINATE

WM\_DDE\_UNADVISE

**Messages traqués par l'option COMMAND USER**

WM\_COMMAND

Messages de WM\_USER à \$7FFF

## **Messages traqués par l'option Presse-papier / impression**

WM\_ASKCBFORMATNAME

WM\_COPY

WM\_DRAWCLIPBOARD

WM\_SIZECLIPBOARD

WM\_HSCROLLCLIPBOARD

WM\_DESTROYCLIPBOARD

WM\_VSCROLLCLIPBOARD

WM\_PASTE

WM\_CUT

WM\_RENDERALLFORMATS

WM\_RENDERFORMAT

WM\_SPOOLERSTATUS

WM\_PAINTCLIPBOARD

WM\_CHANGECHAIN

## **Messages traqués par l'option MENU**

WM\_INITMENU

WM\_INITMENUPOPUP

WM\_MENUCHAR

WM\_MENUSELECT

## **Messages traqués par l'option ICON CURSEUR**

STM\_GETICON

STM\_SETICON

WM\_ICONERASEBKGND

WM\_SETCURSOR

**Messages traqués par l'option CHECK BOX :**

Il s'agit en réalité des messages "boutons" :

BM\_GETCHECK

BM\_GETSTATE

BM\_SETCHECK

BM\_SETSTATE

BM\_SETSTYLE

**Messages traqués par l'option EDIT :**

EM\_CANUNDO

EM\_EMPTYUNDOBUFFER

EM\_FMTLINES

EM\_GETFIRSTVISIBLELINE

EM\_GETHANDLE

EM\_GETLINE

EM\_GETLINECOUNT

EM\_GETMODIFY

EM\_GETPASSWORDCHAR

EM\_GETRECT

EM\_GETSEL

EM\_GETWORDBREAKPROC

EM\_LIMITTEXT

EM\_LINEFROMCHAR

EM\_LINEINDEX

EM\_LINELENGTH

EM\_LINESCROLL

EM\_REPLACESEL

EM\_SETHANDLE

EM\_SETMODIFY

EM\_SETPASSWORDCHAR

EM\_SETREADONLY

EM\_SETRECT

EM\_SETRECTNP

EM\_SETSEL

EM\_SETTABSTOPS

EM\_SETWORDBREAKPROC

EM\_UNDO

WM\_GETTEXT

WM\_GETTEXTLENGTH

WM\_UNDO

WM\_CLEAR

WM\_SETTEXT

**Messages traqués par l'option LIST BOX :**

LB\_ADDSTRING

LB\_DELETESTRING

LB\_DIR

LB\_FINDSTRING

LB\_FINDSTRINGEXACT

LB\_GETCARETINDEX

LB\_GETCOUNT

LB\_GETCURSEL

LB\_GETHORIZONTALEXTENT

LB\_GETITEMDATA

LB\_GETITEMHEIGHT

LB\_GETITEMRECT

LB\_GETSEL

LB\_GETSELCOUNT

LB\_GETSELITEMS

LB\_GETTEXT

LB\_GETTEXTLEN

LB\_GETTOPINDEX

LB\_INSERTSTRING

LB\_RESETCONTENT

LB\_SELECTSTRING

LB\_SELITEMRANGE

LB\_SETCARETINDEX

LB\_SETCOLUMNWIDTH

LB\_SETCURSEL

LB\_SETHORIZONTALEXTENT

LB\_SETITEMDATA

LB\_SETITEMHEIGHT

LB\_SETSEL

LB\_SETTABSTOPS

LB\_SETTOPINDEX

WM\_COMPAREITEM (utilisé également par les combo box)

WM\_DELETEITEM (utilisé également par les combo box)

**Messages traqués par l'option COMBO BOX :**

CB\_ADDSTRING

CB\_DELETESTRING

CB\_DIR

CB\_FINDSTRING

CB\_FINDSTRINGEXACT

CB\_GETCOUNT

CB\_GETCURSEL

CB\_GETDROPPEDCONTROLRECT

CB\_GETDROPPEDSTATE

CB\_GETEDITSEL

CB\_GETEXTENDEDUI

CB\_GETITEMDATA

CB\_GETITEMHEIGHT

CB\_GETLBTEXT

CB\_GETLBTEXTLEN

CB\_INSERTSTRING

CB\_LIMITTEXT

CB\_RESETCONTENT

CB\_SELECTSTRING

CB\_SETCURSEL

CB\_SETEDITSEL

CB\_SETEXTENDEDUI

CB\_SETITEMDATA

CB\_SETITEMHEIGHT

CB\_SHOWDROPDOWN

WM\_COMPAREITEM (idem liste box)

WM\_DELETEITEM (idem liste box)

WM\_CHANGECHAIN

## YR TRACE 3.1

(c) Didier Heiderich 1994, 1995

2, résidence Beauséjour - rue Bach - 38120 Le Fontanil-Cornillon

Aucune distribution commerciale n'est autorisée sans le consentement formel de l'auteur.

Aucune modification ne peut être apportée au logiciel.

Aucun composant du logiciel ne peut être distribué hors de son ensemble, y compris le programme, les bibliothèques dynamiques, la documentation et l'aide.

Le logiciel est à utiliser à vos risques et périls. Il ne pourra en aucun cas être fait état de dommages causés par le logiciel dans votre système ou dans quoi que ce soit.

YRTRACE 3.1 fonctionne selon le principe du "shareware", son utilisation n'est pas libre et doit faire l'objet d'un enregistrement. si vous continuez à l'utiliser plus de 30 jours après sa première utilisation.

Cependant, son enregistrement ne fait l'objet d'AUCUN PAYEMENT (Gratuit).

Pour enregistrer YRTRACE 3.1, écrivez-moi en me fournissant une enveloppe timbrée à votre nom, adresse et raison sociale. Vous recevrez en retour la combinaison de codes à effectuer pour transformer votre version en version enregistrée et débridée. Profitez-en pour me fournir des commentaires (éventuellement sur une carte postale...).

Le nom fourni est le nom réel de l'application qui peut être différent du nom commercial.

Lorsqu'il est connu, c'est le nom du message qui est affiché, sinon le numéro (en autres, pour les messages spécifiques du logiciel).

Le numéro de la fenêtre est le "Handle" (identificateur) HWND de la fenêtre concernée par le message.

Lorsque la fenêtre possède un nom, celui-ci est affiché. Attention : sous Windows, une simple zone de saisie est une fenêtre.

Un message Windows est constitué de 2 paramètres en plus du nom du message :

- **WParam** : un mot de 16 bits
- **LParam** : un mot de 32 bits que YTRACE fourni décomposé en 2 mots de 16 bits  
LParam Lo et LParam Hi en plus du mot complet.

Dans certains cas, LParam est un pointeur sur un texte. Lorsque YR reconnaît un pointeur sur un texte le début du texte est affiché.



Ecriture, lancement et arrêt d'une macro en YRBASIC

## **Le langage**

Introduction

Les séparateurs

Les variables

Les constantes

Les commandes

Les fonctions

Les boucles

Les conditions

Les constantes des utilisateurs

Les messages d'erreur

## **Ecriture, lancement et arrêt d'une macro en YRBasic**

### **Ecriture**

Pour écrire un programme YRBASIC, utilisez un éditeur de texte du type BLOC-NOTES et sauvegardez votre texte avec l'extension ".BYR"

### **Lancement**

Pour lancer un programme YRBASIC, 2 possibilités :

- Choisissez dans le menu : "FICHIER" puis "EXECUTER MACRO", choisissez le fichier BYR à exécuter, cliquez OK et le démarrage est immédiat.

- Associez une macro à un message.

### **Arrêt**

Vous stoppez TOUS les programmes YRBASIC en cours en utilisant les commandes "FICHIER" "STOPPER MACRO", sauf si ces macros sont dans une boucle sans fin et prioritaire (non multitâche).

Exemple de boucles sans fin prioritaire qui bloque tout le système :

```
10 GOTO 10
```

Exemple de boucle sans fin mais multitâche :

```
MULTITASK ON  
10 GOTO 10
```

## Introduction au YRBASIC 1.0

### Pourquoi un basic ?

A l'origine, il n'était pas prévu d'implanter un basic dans YRTRACE. Cependant, comme je voulais que les utilisateurs puissent associer une action à l'arrivée d'un message, le choix du basic s'est imposé de lui même comme système de macro-commandes.

De plus, au départ le pilotage (partiel) de YRTRACE via le Minitel était implanté en dur dans le programme.

Seulement, comme je ne voulais pas aller trop loin dans les possibilités de cette version, j'ai pensé que certains d'entre vous seraient frustrés par les limites de ce pilotage. Aussi, en remplaçant le pilotage en "dur" par un programme basic en parallèle, je limitais mon travail en augmentant les possibilités de YRTRACE, *cuique suum ...*

### Les caractéristiques de YRBASIC

Ne rêvez pas : YRBASIC n'est pas Visual Basic (le prix non plus, puisqu'il est entièrement gratuit) ! Dédié à l'écriture de macro-commandes, il a des **possibilités** :

- \* il est pseudo-multitâche (selon le mode de Windows (\*\*)), et fonctionne en parallèle de YRTRACE (attention : voir la commande : "MULTITASK"),
- \* plusieurs programmes peuvent fonctionner en même temps (sauf en cas de conflits de ressources, par exemple les communications),
- \* il permet d'interagir sur Windows en envoyant des messages aux fenêtres ou aux applications,
- \* il est en relation directe avec certaines fonctions de YRTRACE,
- \* il peut communiquer avec un port série,
- \* vous pouvez écrire en majuscule ou en minuscule sauf les variables.

mais aussi, certaines **limites** :

- \* les programmes sont de 32 Ko de texte au maximum,
- \* vous n'avez pas le droit aux réels, mais à 25 variables alphanumériques (80 caractères max) et 25 variables de type entier long,
- \* il n'y a pas de tableaux,
- \* il n'y a pas de fenêtre dédiée aux programmes YRBASIC : les "print" et autres "input" sont sous forme de fenêtres individuelles type "MessageBox",
- \* il n'y a pas de gestion de fichiers, n'y de gestion de l'imprimante,
- \* les fonctions ne sont pas implantées pour les chaînes de caractères.

Il a des **défauts** :

- pas de contrôle de dépassement des longueurs de chaîne et des nombres : attention, un programme BASIC avec une chaîne de caractère trop longue bloquera le système !!!

**En cas de succès du logiciel (cela dépendra du nombre d'enregistrements que je recevrai...), le YRBASIC sera largement amélioré.**

## Les séparateurs de YRBASIC

Les séparateurs ne peuvent pas être utilisés dans un nom de variable ou de constante

### Calculs

+ : addition  
- : soustraction  
\* : multiplier  
/ : diviser (retourne l'entier)  
^ : puissance  
= : égale  
% : %  
( ) : parenthèses

### Conditions

= : égale  
> : supérieur  
< : inférieur  
! : différent

### Traitement de chaînes de caractères

= : assignation d'une variable alphanumérique  
+ : concaténation d'une chaîne de caractères avec une autre

### Autres

;, < > ! # \$ & ' { } \_ . : [ ]

## Les variables de YRBASIC

YRTRACE ne reconnaît que 2 types de variables :

- les variables numériques qui sont des entiers longs de 32 bits [-2 147 483 648 à 2 147 483 647] nommés A...Z
- les variables alphanumériques de 80 caractères, nommés A\$...Z\$

Vous pouvez cependant utiliser les variables en leur donnant un nom plus long et non réservé, par exemple :

```
BOB=1
```

Attention : si vous nommez une variable BOB et une autre BERNARD, pour YRTRACE ce sera la même variable B.

Exemple1 (EX1.BYR):

```
BERNARD = 1  
PRINT BOB:"BOB=BERNARD"  
END
```

**ATTENTION : il se peut que vous ayez des bogues dus à la latitude laissée aux noms de variables, notamment en cas d'erreur de frappe :**

EX2.BYR

REM En dessous, il y a une erreur de syntaxe non détectée :

```
A=ISTHECOMOPEN  
PRINT A
```

## Les constantes de YRBASIC

YRTRACE a un certain nombre de constantes prédéfinies :

### Divers

TRUE = 1  
FALSE = 0  
ON = 1  
OFF = 0

### Constantes de la fonction "MessageBox"

Résultat de la fonction (selon le bouton poussé dans la boîte de messages) :

IDOK  
IDCANCEL  
IDYES  
IDNO  
IDABORD  
IDRETRY  
IDIGNORE

Icône à l'intérieur de la boîte de messages :

MBICONINFORMATION  
MBICONSTOP  
MBICONQUESTION

Type de boutons dans la boîte de messages :

MBABORTRETRYIGNORE : Réessayer / Abandonner / Ignorer  
MBOKCANCEL : Ok / Abandon  
MBRETRYCANCEL : Réessayer / Abandonner  
MBYESNO : Oui / Non  
MBYESNOCANCEL : Oui / Non / Abandon

### Constantes de la fonction "showWindow"

SWHIDE : cacher  
SWMINIMIZE : Icôniser  
SWRESTORE : normale  
SWSHOW : activer  
SWTOP : toujours au premier plan (toujours visible)  
SWNOTOP : n'est plus toujours au premier plan (toujours visible)

### Constante de YRTRACE

HWINDOW : identificateur de la fenêtre principale de YRTRACE  
HWININSIDE : identificateur de la fenêtre intérieure de YRTRACE  
HTASK : Identificateur de la tâche YRTRACE

## Les commandes de YRBASIC

### Entrées / sorties

INPUT : ouvre une boîte de dialogue pour la saisie d'un nombre ou d'un texte  
PRINT : ouvre une boîte de dialogue contenant les éléments décrits dans print  
YRADDLINE : ajoute un commentaire dans la liste de YRTRACE  
YRCLEARLIST : efface la liste de YRTRACE  
CHOICE : Ouvre une boîte de dialogue avec une boîte de liste permettant de simuler un menu  
BEEP : Avertissement sonore

### Déroulement du programme

GOTO : va au label (N°)  
GOSUB : va au sous-programme de label (N°)  
RETURN : Fin d'un sous-programme  
END : Fin du programme  
MULTITASK : Passage ou non en mode multitâche  
REM : Commentaire dans le programme  
YRRUN : Active / désactive la capture des messages

### Conditions

IF ... THEN : Si (condition) alors (commande)

### Boucles

FOR : Nombre de fois  
NEXT : Fin de boucle

### Variables et traitements de chaînes

CLEARALL : remet à 0 toutes les variables et les déclarations de constantes  
MODIFYSTR : remplace un caractère dans une chaîne par un autre caractère  
CHR : retourne un caractère correspondant à un numéro ASCII  
DECLARE : déclare un nom de constante numérique

### Communication avec le port série

CINPUT : saisie d'un nombre ou d'une chaîne à partir du port série ouvert  
CPRINT : envoi vers le port série ouvert les éléments qui suivent l'instruction  
CGETCHAR : saisie un caractère issu du port série  
OPENCOM : ouvre un port série  
CLOSECOM : ferme le port série ouvert

### Windows

POSTMESSAGE : envoi un message vers une fenêtre sans attendre son traitement  
SENDMESSAGE : envoi un message vers une fenêtre  
POSTAPPMESSAGE : envoi un message vers une application  
SHOWWINDOW : visualise une fenêtre windows selon les options  
GETAPP : retourne les éléments descriptifs d'une application en mémoire  
GETLASTMSG : retourne le dernier message capturé par YRTRACE quelque soit le filtre  
GETLASTFMSG : retourne le dernier message filtré capturé par YRTRACE  
GETMSGNAME : retourne le nom d'un message à partir de son numéro  
GETAPPNAME : retourne le nom d'une application à partir de son identificateur  
SENDDLGITMESSAGE : envoi un message vers un contrôle dans une fenêtre

GETLASTERR

: retourne la dernière erreur détectée par YRTRACE

## Les fonctions de YRBASIC

<u>YESNO</u>	: retourne la réponse à un message Oui / Non
<u>MESSAGEBOX</u>	: provoque une boîte de message et retourne le résultat
<u>ABS</u>	: retourne la valeur absolue d'un nombre
<u>CMPSTRING</u>	: compare 2 chaînes de caractères
<u>RETURNMSGNUM</u>	: retourne le numéro d'un message
<u>LEN</u>	: retourne la longueur d'une chaîne de caractères
<u>EXECPROG</u>	: lance un programme windows et retourne le résultat du lancement
<u>GETVARADDR</u>	: retourne l'adresse d'une variable YRBASIC
<u>GETLO</u>	: retourne le poids faible d'un nombre de 32 bits
<u>GETHI</u>	: retourne le poids fort d'un nombre de 32 bits
<u>MAKELONG</u>	: fabrique et retourne un nombre de 32 bits
<u>GETACTIVEWIN</u>	: retourne l'identificateur HWND de la fenêtre active
<u>ISCOMOPEN</u>	: retourne le drapeau de communication

## INPUT

ouvre une boîte de dialogue pour la saisie d'un nombre ou d'un texte

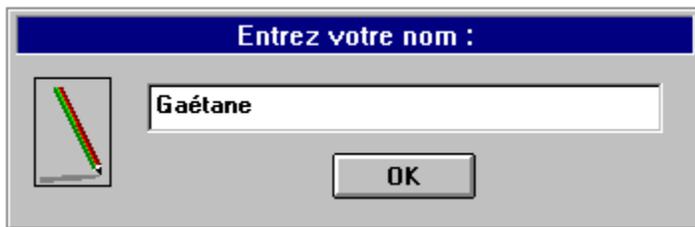
```
INPUT "message",variable
```

"message" est ce qui sera affiché dans la boîte de dialogue (facultatif)  
variable est une variable numérique ou alphanumérique

Exemple :

```
INPUT "Entrez votre nom :",A$  
END
```

Donnera :



## PRINT

ouvre un boîte de dialogue contenant les éléments décrits dans print

PRINT <élément1> <séparateur> <élément2> <séparateur> ..... <élément n> : "titre"

*élément* : variable, une chaîne de caractère ou une expression numérique

*séparateur* : une virgule ou un point virgule. La virgule place le prochain élément à coté, le point virgule provoque une tabulation

"*titre*" : chaîne de caractère qui sera le titre de la boîte de dialogue.

EXEMPLE :

```
INPUT "Entrez votre nom :",A$
```

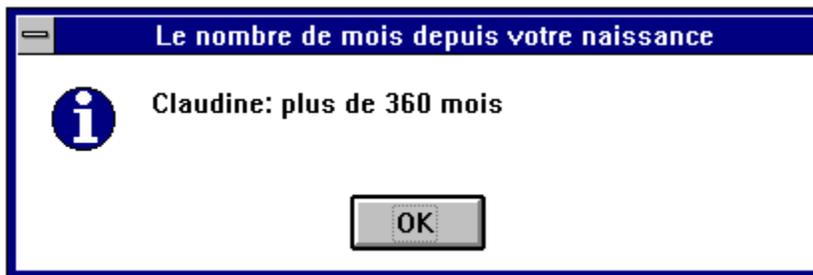
```
INPUT "Entrez votre âge à la fin de cette année:",A
```

```
IF A=0 THEN GOTO 10
```

```
PRINT A$," : plus de ",(A-1)*12, " mois":"Le nombre de mois depuis votre naissance"
```

```
10 END
```

donne :



## YRADDLINE

ajoute un commentaire dans la liste de YRTRACE

YRADDLINE <élément1> <séparateur> <élément2> <séparateur> ..... <élément n>

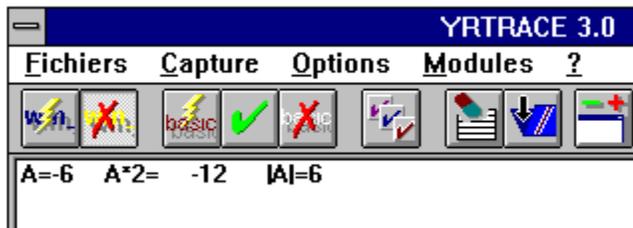
*élément* : variable, une chaîne de caractère, un calcul ou une fonction

*séparateur* : une virgule ou un point virgule. La virgule place le prochain élément à coté, le point virgule provoque une tabulation.

EXEMPLE :

```
INPUT "Entrez un nombre :",A
YRADDLINE "A=",A;"A*2=";A*2;"|A|=",ABS[A]
10 END
```

Résultat :



## CHOICE

Ouvre une boîte de dialogue avec une boîte de liste permettant de simuler un menu

CHOICE <variable>:<argument>:"titre"

*variable* est une variable numérique destinée à recevoir le résultat. Le résultat est le numéro d'ordre de l'item sélectionné (le premier item = 0)

*argument* est une chaîne de caractère ou une variable alphanumérique dont le contenu sont les éléments du menu séparés par des virgules

"*titre*" : est le titre de la boîte placé entre guillemets

Exemple :

```
MULTITASK ON
INPUT "Entrez votre élément de menu : ",A$
IF LEN(A$)=0 THEN A$="(vide)"
B$="BEEP,SALUT,"+A$+",Quitter"
10 CHOICE A:B$:"Votre choix"
IF A=0 THEN BEEP 3
IF A=1 THEN PRINT "Salut"
IF A=2 THEN PRINT A$
IF A!3 THEN GOTO 10
END
```

Résultat :



## **BEEP**

Provoque un signal sonore

BEEP  $n$

$n$  est le nombre de signaux sonores.

## **GOTO**

Fait passer le programme au label qui suit GOTO

GOTO *label*

*label* est un numéro à placer en début de ligne.

Les labels ne peuvent pas être en double

YRBASIC supporte au maximum 100 labels

### EXEMPLE

```
10 INPUT A  
IF A=1 THEN GOTO 10  
END
```

## GOSUB

GOSUB appelle un sous-programme

GOSUB *label*

*label* est un numéro à placer en début de ligne.

Les labels ne peuvent pas être en double.

YRBASIC supporte au maximum 100 labels.

YRTRACE supporte au maximum 50 sous-programmes imbriqués.

Un sous-programme doit impérativement se terminer par RETURN

Exemple :

```
CHOICE A:"Nombre,Nom,Les deux":"Que dois-je entrer"
IF A=0 THEN GOSUB 10
IF A=2 THEN GOSUB 10
IF A=1 THEN GOSUB 20
IF A=0 THEN PRINT B
IF A=1 THEN PRINT B$
IF A=2 THEN PRINT B;B$
END
REM -----
10 INPUT B
IF A!2 THEN RETURN
20 INPUT B$
RETURN
```

## **RETURN**

Met fin à un sous-programme

voir GOSUB

## MULTITASK

Passage ou non en mode multitâche

MULTITASK *interrupteur*

*interrupteur* : 1= Activation 0 = désactivation

Par défaut, le mode est désactivé.

Lorsque le mode multitâche est désactivé, à certaines exceptions (comme l'affichage de boîtes de dialogue), le basic garde la main jusqu'à la fin du programme.

**Le mode multitâche désactivé, une boucle 10 GOTO 10 ne vous rendra jamais la main.**

Aussi, surtout lors de la mise au point de programmes, commencez toujours par l'instruction MULTITASK ON.

Exemple *\_PROG5.BYR* :

```
REM -----
REM           Démonstration de multitask
REM           Exécutez directement cette MACRO
REM           Regardez les boutons de gauche de YRTRACE
REM           en testant les 2 modes.
REM -----

PRINT "Démonstration de la différence entre MULTITASK ON et OFF":YRBASIC"
10 A=YESNO("Multitask ON","")
MULTITASK A
FOR I=1 TO 100
YRRUN ON
YRRUN OFF
NEXT
IF YESNO("Recommencer ?","")=1 THEN GOTO 10
END
```

## **YRRUN**

Active / Désactive la capture des messages et des erreurs selon les options.

YRRUN *interrupteur*

*interrupteur* : 1=activation 0=désactivation

Cette instruction agit directement sur YRTRACE et à le même effet que de sélectionner CAPTURE DEPART / STOP dans le menu de YRTRACE.

Exemple :

```
YRRUN OFF  
END
```

## IF .... THEN

Test conditionnel d'exécution d'une instruction

IF <condition> THEN <instruction>

*condition* : test d'une valeur : > supérieur < inférieur = égale !différent

*instruction* : commande ou assignation

NOTE : le YRBASIC n'accepte pas de condition sur les chaînes de caractères (IF A\$="ESSAI")  
sauf via la fonction CMPSTRING

Exemple :

```
MULTITASK ON
10 INPUT "entrez un nombre :",A
IF A=ABS[A] THEN PRINT "A est un nombre positif"
IF A<0 THEN PRINT "A est un nombre négatif"
IF A=0 THEN A=A+1
IF A=1 THEN GOTO 10
END
```

## FOR / NEXT

FOR et NEXT définissent une boucle à incrémentation

```
FOR <variable numérique> = <valeur de début> TO <valeur de fin>  
<instructions>  
NEXT
```

*variable numérique* : variable utilisée comme compteur de boucle. Cette variable est incrémentée de 1 à chaque passage dans la boucle.

*valeur de début* : valeur de début de la boucle (de la variable)

*valeur de fin* : valeur de la variable mettant fin à la boucle

*instructions* : série d'instructions réalisées dans la boucle

NEXT n'a pas de paramètres, le premier NEXT concerne le dernier FOR

**Le nombre maximum de boucles imbriquées est de 25.**

Exemple :

```
REM ----- N'ENTREZ PAS DE NOMBRE TROP GRAND -----  
MULTITASK ON  
CLEARALL  
10 INPUT "entrez un nombre :",A  
FOR I=1 TO A  
FOR J=I TO A  
X=X+I+J  
BEEP 1  
NEXT  
NEXT  
PRINT "I=",I;"J=",J;"X=",X  
END
```

## MODIFYSTR

Remplace un caractère dans une chaîne par un autre caractère.

MODIFYSTR <variable alphanumérique> ,<caractère à modifier> , <caractère de remplacement>

*variable alphanumérique* : variable contenant la chaîne à modifier et qui recevra en retour la chaîne modifiée.

*caractère à modifier* : variable alphanumérique ou chaîne contenant le caractère à remplacer. Seul le premier caractère est pris en compte.

*caractère de remplacement* :variable alphanumérique ou chaîne contenant le caractère de remplacement. Seul le premier caractère est pris en compte.

exemple :

```
MULTITASK ON
CLEARALL
INPUT "entrez une phrase :",A$
INPUT "entrez le caractère à modifier :",B$
C$=A$
MODIFYSTR C$,B$,"@"
PRINT "<",A$, "> est devenu <", C$, ">"
END
```

## **CHR**

Retourne le caractère correspondant à un numéro ASCII

CHR <numéro>, <variable alphanumérique>

*numéro* : numéro ASCII du caractère à retourner

*variable alphanumérique* : variable de réception du caractère

Exemple :

```
DECLARE "GU"=34
CHR GU,A$
B$=A$+"Guillemets"+A$
PRINT B$
END
```

## **DECLARE**

Déclare un nom de constante numérique

DECLARE "*nom de la constante*" = *valeur de la constante*

*nom de la constante* doit être un nom non réservé par YRTRACE et ne peut pas contenir de séparateur.

*valeur de la constante* : valeur qui sera affectée à la constante

Exemple :

```
DECLARE "AGE"=50  
PRINT "Age = ",AGE  
END
```

## CINPUT

Saisie d'un nombre ou d'une chaîne à partir du port de communication. La fin de la saisie est spécifiée par le caractère ASCII n°13 (retour chariot) qui correspond à la touche "ENTREE".

**La communication doit être établie sinon le système se bloquera**

CINPUT "*message*",*variable*

"*message*" est ce qui sera envoyé au port série (facultatif)

*variable* est une variable numérique ou alphanumérique.

**Attention : si plusieurs programmes utilisent le port série, vous pouvez avoir des conflits. La situation est à éviter.**

Exemple de programme de communication

## **CPRINT**

Envoi vers le port série les éléments décrits dans print

**La communication doit être établie sinon le système se bloquera**

CPRINT <élément1> <séparateur> <élément2> <séparateur> ..... <élément n>

*élément* : variable, une chaîne de caractère ou une expression numérique

*séparateur* : une virgule ou un point virgule. La virgule place le prochain élément à côté, le point virgule provoque une tabulation.

Si CPRINT se termine par une virgule ou un point virgule, YRBASIC ne passera pas une ligne sur le terminal.

Exemple de programme de communication

## **CGETCHAR**

Saisie un caractère issu du port série

**La communication doit être établie sinon le système se bloquera**

CGETCHAR <variable numérique>

*variable numérique* : variable qui récupère le numéro du caractère.

Si le résultat = -1 : aucun caractère n'est disponible

**Attention : si plusieurs programmes utilisent le port série, vous pouvez avoir des conflits. La situation est à éviter.**

Exemple de programme de communication

## **OPENCOM**

Ouvre une communication avec un port série.

COPEN "*chaîne d'initialisation*"

*chaîne d'initialisation* : arguments identiques au DOS (*COMX:VVVVV,P,DB,BS*)

**Attention : cette instruction agit directement sur les communications de YRTRACE.**

Dans son ensemble, YRTRACE ne gère qu'un seul port série, de fait l'ouverture, la fermeture, l'écriture, etc... sur un port va modifier l'ensemble des communications dans les programmes BASIC et dans YRTRACE.

Exemple de programme de communication

## **CLOSECOM**

Ferme la communication avec un port série.

**Attention : cette instruction agit directement sur les communications de YRTRACE.**

Dans son ensemble, YRTRACE ne gère qu'un seul port série, de fait l'ouverture, la fermeture, l'écriture, etc... sur un port va modifier l'ensemble des communications dans les programmes BASIC et dans YRTRACE.

Exemple de programme de communication

## Exemple de programme de communication

Z=ISCOMOPEN

IF Z=1 THEN GOTO 5 REM Test si la communication est déjà établie

CHOICE A:"COM1,COM2,Abandon": "Port série"

IF A=0 THEN COPEN "COM1:1200,E,7,1"

IF A=1 THEN COPEN "COM2:1200,E,7,1"

IF A>1 THEN END

5 CPRINT ""

CPRINT ""

CPRINT "Votre choix"

CPRINT "1.....Entrez une valeur"

CPRINT "2.....Entrez un texte"

CPRINT "3.....Quitter"

10 CGETCHAR A

IF A<0 THEN GOTO 10

IF A=49 THEN GOSUB 100

IF A=50 THEN GOSUB 110

IF A=51 THEN GOTO 20

GOTO 5

20 IF Z!1 THEN CCLOSE

PRINT "Au revoir"

END

100 REM =====

CPRINT ""

CINPUT "Entrez une valeur :",E

CPRINT "Vous avez saisi ",E

RETURN

110 REM =====

CPRINT ""

CINPUT "Entrez un texte :",E\$

CPRINT "Vous avez saisi ",E\$

RETURN

## POSTMESSAGE / SENDMESSAGE

POSTMESSAGE :

Envoi un message vers une fenêtre sans attendre son traitement. C'est la fonction *PostMessage* de Windows.

SENDMESSAGE :

Envoi un message vers une fenêtre et attend le traitement du message . C'est la fonction *SendMessage* de Windows. **Préférez POSTMESSAGE à cette fonction sauf en connaissance de cause.**

POSTMESSAGE <HWND>,<n° du message>,<WPARAM>,<LPARAM>

SENDMESSAGE <HWND>,<n° du message>,<WPARAM>,<LPARAM>

*HWND* :Identificateur de la fenêtre. Voir :

- HWINDOW (constante de YRBASIC)
- HWININSIDE (constante de YRBASIC)
- GETLASTMSG
- GETLASTFMSG
- GETACTIVEWIN

*n° du message* : le message via son numéro. Vous pouvez obtenir le numéro d'un message via RETURNMSGNUM.

*WPARAM* et *LPARAM* : Paramètres classiques des messages Windows : voir MAKELONGX, MAKELONG, GETHI, GETLO, GETVARADDR

Exemple :

```
MULTITASK ON
CLEARALL
A=EXECPROG("WRITE.EXE")
IF A<31 THEN END
10 CHOICE A:"OUVRIR UN FICHER,POLICE,QUITTER": "WRITE"
IF A=0 THEN POSTMESSAGE GETACTIVEWIN,RETURNMSGNUM("WM_COMMAND"),4098,0
IF A=1 THEN POSTMESSAGE GETACTIVEWIN,RETURNMSGNUM("WM_COMMAND"),4168,0
IF A=2 THEN POSTMESSAGE GETACTIVEWIN,RETURNMSGNUM("WM_QUIT"),0,0
IF A<2 THEN GOTO 10
END
```

## POSTAPPMESSAGE

Poste un message à une application.

POSTAPPMESSAGE <HTASK>,<n° du message>,<WPARAM>,<LPARAM>

HTASK :Identificateur de l'application. Voir :

- HTASK (constante de YRBASIC)
- GETLASTMSG
- GETLASTFMSG
- GETAPP

n° du message : le message via son numéro. Vous pouvez obtenir le numéro d'un message via RETURNMSGNUM.

WPARAM et LPARAM : Paramètres classiques des messages Windows : voir MAKELONGX, MAKELONG, GETHI, GETLO, GETVARADDR

Exemple :  
MULTITASK ON  
INPUT "Numéro du message :",A  
POSTAPPMESSAGE HTASK,A,0,0  
END

## SHOWWINDOW

Visualise une fenêtre windows selon les attributs

SHOWWINDOW <Attribut>

*Attribut* : nombre définissant l'attribut d'une fenêtre

Constantes d'attributs :

SWHIDE : cacher  
SWMINIMIZE : Icôniser  
SWRESTORE : normale  
SWSHOW : activer  
SWTOP : toujours au premier plan (toujours visible)  
SWNOTOP : n'est plus toujours au premier plan (toujours visible)

Exemple :

```
MULTITASK ON
SHOWWINDOW HWINDOW,SWTOP
FOR I=1 TO 3
SHOWWINDOW HWINDOW,SWMINIMIZE
BEEP 1
SHOWWINDOW HWINDOW,SWRESTORE
NEXT
END
```

## GETAPP

Retourne les éléments descriptifs d'une application en mémoire

GETAPP <numéro>, <variable numérique>, <variable alphanumérique>

*numéro* : numéro d'ordre de l'application que vous fournissez.

*variable numérique* : variable qui récupère le HTASK de l'application (0 si aucune application ne correspond au numéro)

*variable alphanumérique* : variable qui récupère le nom de l'application.

Exemple :

```
MULTITASK ON
CLEARALL
I=1
B$=""
10 GETAPP I,A,A$
I=I+1
B$=B$+A$
IF LEN(A$)>0 THEN B$=B$+", "
IF A!0 THEN GOTO 10

CHOICE C:B$:"Fermer une application"
GETAPP C+1,A,B$
POSTAPPMESSAGE A,RETURNMSGNUM("WM_QUIT"),0,0
END
```

## GETLASTMSG / GETLASTFMSG

GETLASTMSG : retourne le dernier message capturé par YRTRACE

GETLASTFMSG : retourne le dernier message capturé et filtré par YRTRACE

GETLASTMSG <*hwindow*>,<*htask*>,<*message*>,<*wparam*>,<*lparam*>,<*nom fenêtre*>,<*classe*>

GETLASTFMSG <*hwindow*>,<*htask*>,<*message*>,<*wparam*>,<*lparam*>,<*nom fenêtre*>,<*classe*>

*hwindow* : variable numérique qui récupère le HWND du message

*htask* : variable numérique qui récupère le HTASK du message

*message* : variable numérique qui récupère le message

*wparam* : variable numérique qui récupère le WPARAM

*lparam* : variable numérique qui récupère le LPARAM

*nom fenêtre* : variable alphanumérique qui récupère le nom de la fenêtre

*classe* : variable alphanumérique qui récupère le nom de la classe de la fenêtre

Exemple :

```
GETLASTFMSG A,B,C,D,E,F$,G$
```

```
GETMSGNAME C,A$
```

```
YRADDLINE "##### Message:",A$;"WIN:",F$;"CLASS:",G$,"#####"
```

```
BEEP 1
```

```
END
```

## GETMSGNAME

Retourne le nom d'un message à partir de son numéro

GETMSGNAME <numéro>, <nom>

*numéro* : numéro du message

*nom* : variable alphanumérique destinée à récupérer le nom du message.

Exemple :

```
MULTITASK ON
CLEARALL
D=GETMSGNUM("WM_COMMAND")
FOR I=D TO D+10 REM : Attention : les dépassements de longueur de chaîne ne sont pas
vérifiés !!!
GETMSGNAME I,A$
B$=B$+A$+" ,"
NEXT
CHOICE A:B$:"Messages"
END
```

## **GETAPPNAME**

Retourne le nom d'une application à partir de son identificateur HTASK

GETAPPNAME <*htask*>,<*nom*>

*htask* : identificateur qui vous fournissez

*nom* : variable alphanumérique qui récupère le nom

Exemple :

```
MULTITASK ON  
CLEARALL  
GETAPPNAME HTASK,A$  
PRINT A$:"Mon nom"  
END
```

## SENDDLGITEMMESSAGE

Envoi un message vers un contrôle dans une fenêtre. La fonction est identique à celle de Windows.

SENDDLGITEMMESSAGE <HWND>,<ID>,<MESSAGE>,<LPARAM>,<WPARAM>

*HWND* :Identificateur de la fenêtre. Voir :

- HWINDOW (constante de YRBASIC)
- HWININSIDE (constante de YRBASIC)
- GETLASTMSG
- GETLASTFMSG
- GETACTIVEWIN

*ID* : Identificateur du contrôle

*MESSAGE* : le message via son numéro. Vous pouvez obtenir le numéro d'un message via RETURNMSGNUM.

*WPARAM et LPARAM* : Paramètres classiques de messages Windows : voir MAKELONGX\_MAKELONG, GETHI, GETLO, GETVARADDR

Exemple :

```
MULTITASK ON
CLEARALL
A$="The full mind is alone the clear (Schiller)"
SENDDLGITEMMESSAGE HWININSIDE,400,RETURNMSGNUM("LB_ADDSTRING"),0,GETVARADDR(A$)
END
```

## **GETLASTERR**

Retourne la dernière erreur détectée par YRTRACE

GETLASTERR <*variable alphanumérique*>

*variable alphanumérique* : variable qui récupère la dernière erreur produite dans une application

Exemple :

```
MULTITASK ON  
CLEARALL  
GETLASTERR A$  
PRINT A$  
END
```

## YESNO (fonction)

Renvoi la réponse à un message Oui / Non fourni dans une boîte de dialogue

YESNO(<commentaire>,<titre> )

**En retour** : 1 ou 0 (oui / non)

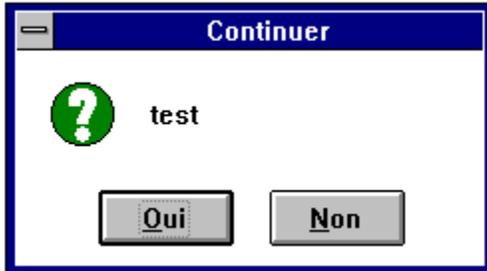
*commentaire* : chaîne ou variable alphanumérique s'affichant dans la boîte

*titre* : chaîne ou variable alphanumérique du titre de la boîte

EXEMPLE :

```
10 IF YESNO("test","Continuer")=TRUE THEN GOTO 10  
END
```

Résultat :



## MESSAGEBOX (fonction)

provoque une boîte de message et retourne le résultat.

MESSAGEBOX[<commentaire>,<titre> ,<icône>,<bouton>]

**En retour** : réponse du bouton pressé.  
nombres prédéfinis dans des constantes YRBASIC:

IDOK  
IDCANCEL  
IDYES  
IDNO  
IDABORD  
IDRETRY  
IDIGNORE

*commentaire* : chaîne ou variable alphanumérique s'affichant dans la boîte

*titre* : chaîne ou variable alphanumérique du titre de la boîte

*icône* : affichage d'une icône dans le boîte de dialogue, selon les valeurs suivantes :

MBICONINFORMATION : le " i " d'information  
MBICONSTOP : stop  
MBICONQUESTION : point d'interrogation

*bouton* : types de boutons dans la boîte de dialogue :

MBABORTRETRYIGNORE : Réessayer / Abandonner / Ignorer  
MBOKCANCEL : Ok / Abandon  
MBRETRYCANCEL : Réessayer / Abandonner  
MBYESNO : Oui / Non  
MBYESNOCANCEL : Oui / Non / Abandon

Exemple :

```
MULTITASK ON
CLEARALL
A$="Selon vous, la finalité est l'objectif du finalisme ?"
A=MESSAGEBOX[A$,"TEST DE COMPREHENSION",MBICONQUESTION,MBYESNOCANCEL]
END
```

## **ABS (fonction)**

Retourne la valeur absolue d'un nombre

ABS[<expression numérique>]

**En retour** : La valeur absolue de l'expression numérique

Exemple :

```
MULTITASK ON  
CLEARALL  
INPUT A  
PRINT -ABS[-1*A]  
END
```

## **CMPSTRING (fonction)**

compare 2 chaînes de caractères.

CMPSTRING(<chaîne ou variable alphanumérique>,<chaîne ou variable alphanumérique>)

**En retour :** 0 : les chaînes sont différentes  
1 : les chaînes sont égales  
2 : les chaînes sont égales si l'on fait abstraction des majuscules / minuscules

Exemple :

```
MULTITASK ON
INPUT "Entrez le mot : 'ACIDE DESOXYRIBONUCLEIQUE' ",A$
IF CMPSTRING(A$,"ACIDE DESOXYRIBONUCLEIQUE")>0 THEN PRINT "Et maintenant la formule ?"
END
```

## **RETURNMSGNUM (fonction)**

Retourne le numéro d'un message.

RETURNMSGNUM(<chaîne ou variable alphanumérique>)

**En retour** : Le numéro du message contenu dans <chaîne ou variable alphanumérique> ou 0 si le message n'est pas répertorié par YRTRACE.  
La fonction n'accepte pas des expressions du type "WM\_USER+10" !

voir aussi :

POSTMESSAGESENDMESSAGEPOSTAPPMESSAGE

Exemple :

```
MULTITASK ON
INPUT "Message :",A$
PRINT A$,"=",RETURNMSGNUM(A$)
END
```

## **LEN (fonction)**

Retourne la longueur d'une chaîne de caractères.

LEN(<chaîne ou variable alphanumérique>)

**Retour :** la longueur de <*chaîne ou variable alphanumérique*>

Exemple :

```
A$="123456789"  
PRINT LEN(A$)  
END
```

## **EXECPROG (fonction)**

Lance un programme windows et retourne le résultat du lancement.

EXECPROG(<*chemin\programme*>)

**Retour :** Un nombre > 31 si le lancement du programme est effectué sans problème, sinon la valeur de retour de *WINEXEC* (voir documentation Microsoft).

*chemin\programme* : le chemin et le nom dos de l'application Windows.

Exemple :

```
EXECPROG("WRITE.EXE")  
END
```

## **GETVARADDR (fonction)**

Retourne l'adresse d'une variable YRBASIC

GETVARADDR(<variable>)

**Retour :** l'adresse de la variable <variable>

Exemple :

```
MULTITASK ON
CLEARALL
A$="Les sanglots longs des violons de l'automne bercent mon coeur d'une langueur monotone
(Verlaine)."
SENDDLGITMESSAGE HWININSIDE,400,RETURNMSGNUM("LB_ADDSTRING"),0,GETVARADDR(A$)
END
```

## **GETLO / GETHI (fonctions)**

GETLO[*<expression numérique>*] : retourne le poids faible d'un nombre de 32 bits

GETHI[*<expression numérique>*] : retourne le poids fort d'un nombre de 32 bits

Utile pour traiter certains messages de Windows.

**Retour** : la valeur correspondante

Exemple :

```
A=144000  
PRINT "A=",A;"LO(A)=",GETLO[A];"HI(A)=",GETHI[A]  
END
```

## **MAKELONG (fonction)**

Fabrique et retourne un nombre de 32 bits

MAKELONG[<mot de poids faible>,<mot de poids fort>]

**Retour :** un mot de 32 bits fabriqué à partir de <mot de poids faible>,<mot de poids fort>

Exemple :

```
A=144000
B=GETLO[A]
C=GETHI[A]
E=MAKELONG(B,C)
PRINT "A=",A;"LO(A)=",GETLO[A];"HI(A)=",GETHI[A];" Make long=",E
END
```

## **GETACTIVEWIN (fonction)**

Retourne l'identificateur HWND de la fenêtre active.

**Retour :** le HWND de la fenêtre active.

Exemple :

```
MULTITASK ON
CLEARALL
A=EXECPROG("WRITE.EXE")
IF A<32 THEN END
10 CHOICE A:"OUVRIR UN FICHIER,POLICE,QUITTER": "WRITE"
IF A=0 THEN POSTMESSAGE GETACTIVEWIN,RETURNMSGNUM("WM_COMMAND"),4098,0
IF A=1 THEN POSTMESSAGE GETACTIVEWIN,RETURNMSGNUM("WM_COMMAND"),4168,0
IF A=2 THEN POSTMESSAGE GETACTIVEWIN,RETURNMSGNUM("WM_QUIT"),0,0
IF A<2 THEN GOTO 10
END
```

## **ISCOMOPEN (fonction)**

Retourne le drapeau de communication

**Retour :** 0 si absence de communication, sinon 1

Exemple :

```
IF ISCOMOPEN=0 THEN PRINT "Pas de communication en cours"  
END
```

## Les erreurs dans le YRBASIC

Le YRBASIC affiche les erreurs dans une boîte de dialogue, avec l'erreur dans le titre et l'emplacement de l'erreur dans la boîte.

Titre : on retrouve dans l'ordre :

l'élément ayant provoqué l'erreur : un commentaire supplémentaire sur l'erreur (en général la fonction) : le type d'erreur.

### Types d'erreur :

"Erreur de syntaxe" : l'expression ou l'instruction n'est pas correcte

"Erreur de parenthèses" : absence de parenthèse ou de guillemets, ou présence non attendue d'un séparateur

"Absence d'expression" : absence d'une expression attendue

"Erreur sur une variable" : la variable n'est pas correcte

"Table des labels complète (100 max)" : vous avez défini trop de labels

"Label en double" : le label existe déjà

"Label inconnu" : le label n'existe pas (GOTO ou GOSUB)

"Absence de THEN" : après une condition dans un IF, vous devez mettre un THEN

"Absence de TO" : absence de TO dans une boucle FOR

"Trop de boucles FOR" : vous avez dépassé le nombre maximum de boucles imbriquées

"NEXT sans FOR" : la boucle n'existe pas ou est déjà fermée

"Trop de GOSUB" : dépassement du nombre maximum de GOSUB

"RETURN sans GOSUB préalable" : ce n'est pas un sous-programme

"Table des constantes pleines (50 max)" : trop de constantes sont définies

"Erreur communication" : la communication n'est pas établie

"Dépassement d'un nombre" : dépassement de capacité

### Emplacement de l'erreur

Attention si l'erreur est en bout de ligne, YRTRACE vous affichera la ligne suivante pour indiquer l'emplacement de l'erreur.

## Je cherche un job (mi-95) !

Je cherche un job. 4KTrace est un échantillon de mes compétences. Si vous voulez en savoir plus que le CV qui suit, je suis à votre disposition.  
D. Heiderich.

### Didier Heiderich - Ingénieur cesi

2, résidence Beauséjour - rue Bach - 38120 Le Fontanil-Cornillon (Grenoble)  
31 ans, né à Metz (Moselle), marié, un enfant (18 mois).

### Etudes...

1990 - 1992 : **INGENIEUR CESI** Centre d'Etudes Supérieures Industrielles à Ecully (69)  
*Formation d'ingénieur généraliste - Spécialisation : informatique.*

1984 : Technicien Supérieur Physicien Chimiste à l'AFPA de Champs S/Marne (77)

### Expérience...

1993 - 1994: **Ingénieur logiciel** pour JMBS (38)

*SSI en informatique industrielle et technique de 4 personnes.*

- Réalisation de logiciels en C, C++, Pascal O.O sous DOS et Windows (projets de 300 à 400 KF).  
Recensement des besoins, réalisation des cahiers des charges, définition des systèmes, développement, maintenance et formation des utilisateurs.

*Résultats: doublement du C.A et attaque du marché américain.*

1992 : **Ingénieur Système d'Information et Gestion** pour ALR (69)

*Mécanique / plasturgie, 2 sites, 180-200 personnes en production - Coût du projet: 800 KF*

- Création et adaptation d'un système de production du type mixte MRP / JAT,

- Mise en place de la bureautique en production, formation des utilisateurs,

- Première phase de la mise en place de la qualité (ISO9002).

Etat des lieux, proposition et définition du projet avec la direction, sensibilisation

du personnel, relation avec les clients, création et animation des groupes de projet, formation des responsables de production, mise en place et adaptation des systèmes, bilan final.

*Résultats: Augmentation de 12 à 15 % de la productivité et obtention de nouveaux marchés.*

1991 : **Mission en Organisation Informatique** pour CARRIER S.A (01)

*Climatisation / Froid, 3 sites, Service Achats: 15000 références / 1000 fournisseurs / 20000 commandes/an.*

- Dans le cadre de la réorganisation de la gestion de production (Just In Time) de l'ensemble de

l'activité France, définition du système d'information du service Achats.  
Etudes des besoins, analyse fonctionnelle, définition, jonction au projet global.  
*Résultats: Réalisation d'un cahier des charges complètement intégré à l'ensemble du projet.*

**1986 - 1990 : Informaticien-Chimiste pour Rhône-Poulenc Agro (69)**

*Laboratoire d'études et de recherche - projets de 6 mois à 2 ans*

- Correspondant informatique,

formation des utilisateurs, aide technique, installation (+30 personnes).

- Développements de logiciels complets et distribués, utilitaires, routines pour la robotique

*Résultats: 1 logiciel distribué dans l'ensemble du groupe Rhône-Poulenc (monde) et*

*1 logiciel distribué dans le groupe et à l'extérieur (INRA, CEA, groupes industriels...),*

- Réalisation d'études échochimiques d'homologation pour l'EPA (USA), l'OCDE, la BBA (RFA),

- Mise au point de méthodes d'analyse des produits nouveaux,

Définitions, protocoles, réalisation des études, dossier pour les instances internationales,

*Résultats: Obtention de toutes les homologations dans tous les délais programmés.*

**1985 : Formateur à l'AFPA de Champs S/Marne (77)**

- Formation en des élèves Techniciens Supérieurs Physiciens Chimistes.

*formation en physique et en informatique théorique et pratique.*

1983 : Ouvrier de Production pour Diamond Shamrock France à Courtenay (45).

## **Divers...**

Anglais courant: séjours aux USA et langue de travail (RPA et JMBS).

A titre personnel, réalisation de logiciels (DOS, WINDOWS) distribués sous diverses formes.

