

WM_COMMAND LB GET
M_GETSTATE err FAILUR
WINDOW GDI_FREE STM
CD_STRING IF A=3 T
MESSAGE CLASS BO
OPENCOM WM_NCLBITT

YRTRACE
Départ
Stop

YRBASIC 1.0

Sommaire

Que fait YRTRACE3
Configuration et installation4
Quoi de nouveau? et limites de la version shareware...5
Options6
YRTRACE et les communications sur les ports séries ...7
Capture des messages8
Sélection des applications8
Sélection des messages8
Définir visualisation9
Associer Macro9
Les erreurs détectées10
Liaison avec vos logiciels.....11

Le YRBASIC12
Ecriture, lancement et arrêt d'une macro en YRBASIC ..13
Les séparateurs14
Les variables14
Les constantes15
Les commandes et instructions15
Les fonctions32
Les erreurs37
Index du YRBASIC38

Je cherche un job40

YRTRACE 3.1 / YRBASIC 1.0

(c) Didier Heiderich 1994, 1995

2, résidence Beauséjour - rue Bach - 38120 Le Fontanil-Cornillon (FRANCE)

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. Les nouvelles fonctions de la version 3.1 sont légèrement bridées.

Cependant, **son enregistrement ne fait l'objet d'AUCUN PAYEMENT (Gratuit)**, il a pour but d'apporter des informations pour améliorer le logiciel et également pour savoir s'il est utile de continuer d'améliorer et de distribuer YRTRACE, du moins en version française.

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 complète enregistrée. Profitez-en pour me fournir des commentaires (éventuellement sur une carte postale...).

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,

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 !

L'aide complète est fournie avec le logiciel, c'est pourquoi ce document est limité aux parties les plus utiles.

Configuration et installation

Configuration

Il vous faut

- au minimum un 386 avec 2Mo de mémoire et un écran couleur à partir d'EGA.
- Windows 3.1 ou 3.11

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_bc.cpp.exe
ex_link.by
plus le code source de la démonstration..

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

Installation

Copier tous les fichiers sous un répertoire YRTRACE puis ajouter classiquement les programmes YRTRACE.EXE et EX_BC.cpp.EXE via le gestionnaire de programmes. Lors de la première utilisation de YRTRACE, l'installation se fait automatiquement. Tous les fichiers restent dans le répertoire d'origine sauf CTL3DV2.____ qui est copié dans votre répertoire Windows sous le nom CTL3DV2.DLL (*) s'il n'est pas déjà présent.

(*) Windows et CTL3DV2.DLL sont des produits (c) Microsoft Corporation.

Quoi de nouveau? et limites de la version 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 notepad 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,

- (#) pour améliorer ce principe, des instructions permettent de déclencher et stopper 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. Voir page 3.

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!

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 téléinformatique de 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.

Capture des messages

Départ provoque la capture, de même que l'instruction YRBASIC "YRRUN ON", et ceci en fonction des options.

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"

Sélection des applications

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

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.

Sélection des messages

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

Classification des messages

Les messages sont répertoriés dans YRTRACE selon 18 classes :

- Edition : messages concernant l'édition de texte.
- Boîtes de listes : messages utilisés par les boîtes de liste.
- Icône curseur : messages concernant les icônes et les curseurs
- Cases à cocher : messages des 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
- Multimédia : messages multimédias
- Autres : messages non classés
- Utilisateurs : messages définis par les utilisateurs (entre WM_USER et \$7FFF)
- Non répertoriés : messages non répertoriés (YRTRACE fourni le n° du message).

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 : ADDLINE et YRCLEARLIST

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

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.

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).

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".

Liaison avec vos programmes

Vous pouvez créer un lien entre YRTRACE et vos programmes 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-ci 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

Note: "(#)" désigne des options uniquement utilisables après enregistrement. Voir page 3

Le 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 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.

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 : "FICHER" 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 "FICHER" / "STOPPER MACRO", sauf si ces macro 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
```

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 [de -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 dues 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

Constante de YRTRACE

HWINDOW : identificateur de la fenêtre principale de YRTRACE
HWININSIDE : identificateur de la fenêtre intérieur 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 un 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 quel que 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

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

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
```

YRADDLINE

ajoute un commentaire dans la liste de YRTRACE

```
YRADDLINE <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 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
```

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
```

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 a 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, 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 paramètre.

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.

voir : exemple de programme de communication

CPRINT

Envoie 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 coté, 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.

voir : 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.

voir : 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.

voir : 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.

voir : 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 d'un message Windows : voir MAKELONG, GETHI, GETLO, GETVARADDR

Exemple :

MULTITASK ON

CLEARALL

A=EXECPROG("WRITE.EXE")

IF A<31 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

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 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	: icôner
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 aux dépassements de longueur de chaîne!!!
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
```

SENDDLITEMESSAGE

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

SENDDLITEMESSAGE <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 MAKELONG, GETHI, GETLO, GETVARADDR

Exemple :

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

Les fonctions de YRBASIC

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

YESNO (fonction)

Renvoie 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
```

MESSAGEBOX(fonction)

provoque une boîte de message et retourne le résultat.

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

En retour : réponse du bouton pressé.

nombre prédéfinis dans des constantes YRBASIC:

```
IDOK  
IDCANCEL  
IDYES  
IDNO  
IDABORT  
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 capitales

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 les expressions du type "WM_USER+10".

voir aussi :

POSTMESSAGE, SENDMESSAGE, POSTAPPMESSAGE

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$="Les sanglots long des violons de l'automne bercent mon coeur d'une langueur monotone (Verlaine)."
SENDLGITEMMESSAGE HWININSIDE,400,RETURNMSGNUM("LB_ADDSTRING"),0,GETVARADDR(A$)
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 définis 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.

YRTRACE provoque une erreur dans Windows

Vous dépassez le maximum de 80 caractères dans une expression alphanumérique,
Vous dépassez la capacité de calcul,
Vous utilisez un port série qui n'est pas ouvert ou qui est invalide.

Index du YRBASIC

A

abs 33

B

beep 18

C

cgetchar 24

choice 17

cinput 24

clearall 16

cmpstring 33

closecom 25

commandes du yrbasic 15

constantes du yrbasic 14

cprint 24

chr 23

D

declare 23

E

end 15

erreurs (messages d'erreurs) 37

execprog 34

exemple de programme de communication 26

F

false (constante prédéfinie) 14

fonctions (les fonction du yrbasic) 32

for 22

G

getactivewin 36

getapp 29

getappname 30

gethi 35

getlasterr 31

getlastfmsg 29

getlastmsg 29

getlo 35

getmsgname 30

getvaraddr 35

gosub 18

goto 18

H

htask 15

hwininside 15

hwindow 15

I

idok, idcancel, idyes, idno, idabord, idretry, idignore (retour de MESSAGEBOX) : 14, 32

if 21

input 16

iscomopen 36

L

len 34
limites du yrbasic 12

M

makelong 35
mbiconinformation, mbiconquestion, mbiconstop (icône de MESSAGEBOX) : 14, 32
mbabordretryignore, mbokcancel, mbyesno,
mbyesnocancel (boutons de MESSAGEBOX): 14, 32
messagebox 32
modifystr 22
multitask 20

N

next 22

O

off (constante prédéfinie) 14
on (constante prédéfinie) 14
opencom 25

P

postappmessage 28
postmessage 27
print 17

R

return 19
returnmsgnum 34

S

senddlgitemmessage 31
sendmessage 27
séparateurs dans le yrbasic 12
showwindow 28
swhide, swminimize, swrestore, swshow, swtop, swnotop : 15, 28

T

then 21
true (constante prédéfinie) 14

V

variables 14

Y

yesno 32
yraddline 17
yrclearlist 15
yrrun 20

Je cherche un job. 4KTrace
est un échantillon de mes compé-
tences. 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 : organisation et 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).

formation final.

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, des responsables de production, mise en place et adaptation des systèmes, bilan

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.

Achats.

- 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

Etudes des besoins, analyse fonctionnelle, définition, jonction au projet global.

projet.

Résultats: Réalisation d'un cahier des charges complètement intégré à l'ensemble du

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,
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.

Vous pouvez également consulter le CV hypertexte plus complet à partir de YRTRACE [CVDH.HLP].