

Contents

Start Up

Starting Thoroughbred <i>Basic</i>	1
For More Information	1
Directives	2
Functions & System Variables	16
<i>Dictionary-IV</i> APIs	29
I/O MNEMONICS	31
Additional Mnemonics	34
ASCII Codes	35
Error Codes	36
System	36
Input/Output	36
File Usage	37
Structure	37
Limit	37
Execution	37
Transaction Processing	38
Windows	38
Other	39
<i>Dictionary-IV</i> Interface	39
Notational Symbols	41
Software Conventions	43
Function Keys	43
Function Keys from the Definition Name Field	43
Cursor Movement During Field Edit	43
Types of Menus	45
Using IDOL-IV Menus	45
Printing	47
Common Syntax Elements	47
Mask Characters (Numeric Output)	52
Date Masking Characters	52
Services	53
Printed Documentation	53
On-line Documentation	53
Support	53
On-line Communication	54
Training Classes	55

Thoroughbred *Basic* Quick Reference

(8.3.1 and up)

Starting Thoroughbred *Basic*

From DOS change to the directory that holds B.EXE (the default is TBSC). Type **B** and press **Enter**.

From the IDOL-IV Control Menu. Select option **9 Basic Utilities**.

From the Thoroughbred *Basic* Utilities Menu select option **22 BASIC**.

Note: To return to IDOL-IV from Thoroughbred *Basic* type **RUN"ID"** and press **Enter**.

For More Information

...on 3GL system functions and system variables, refer to the Thoroughbred *Basic* Reference Manuals, Thoroughbred *Basic* Utilities Manual, and Thoroughbred *Basic* Customization and Tuning Guide

...on *Dictionary-IV* features, refer to the Thoroughbred *Dictionary-IV* Reference Manuals.

...on *Query-IV* features, refer to the Thoroughbred *Query-IV* Reference Manual.

...on *Report-IV* features, refer to the Thoroughbred *Report-IV* Reference Manual.

...on *Script-IV* features, refer to the Thoroughbred *Script-IV* Reference Manual.

Directives

A release level number preceding the syntax indicates availability starting at that release.

ADD - Add Filename

```
ADD file-name[,ERR=line-ref]
```

ADDR - Add Memory Resident Public Program

```
ADDR program-name[,ERR=line-ref ]
[,BNK=bank-num]
```

ADDSORT - Add New Sort Sequence

```
8.1 ADDSORT file-name,[sort-name1:] sortdef1
[:mode1] [,[sort-name2:] sortdef2
[:mode2] [...[sort-namen:] sortdefn
[:moden]]],disk-num, [,ERR=line-ref]
```

API - Interface To The Microsoft Windows API

```
8.3.1 API(library$,function$[,argument-1...
argument-n])
```

BEGIN - Begin Program Environment

```
BEGIN [,EXCEPT variable-name
[,variable-name...]]
```

CALL - Call Public Program

```
CALL program-name [,ERR=line-ref] [,value-
list]
```

CLEAR - Clear Program Environment

```
CLEAR [,EXCEPT variable-name
[,variable-name...]]
```

CLEAR ERC - Clear Error Condition Variable

```
8.3 CLEAR ERC
```

CLOSE - End I/O Channel Operations

```
CLOSE (channel [,ERR=line-ref])
```

COMMIT - End Tracking Record Changes and Make I/O Changes Permanent

```
8.2 COMMIT [,ERR=line-ref]
```

DEF FN - Define Function

```
DEF FN x[$] (variable-list) =
numeric/string expression
```

DELETE - Delete Program Statements

```
DELETE [line-ref1 [,line-ref2]]
or
DELETE [,line-ref2]
```

DELETE ARRAY - Delete Elements Of An Array

```
3.2 DELETE ARRAY array-name [(pos1,count1)
[, (pos2,count2)[, (pos3,count3)]]]
```

DIM - Dimension Numeric Array

```
DIM array-name (dim1 [,dim2 [,dim3]])
[,array-name (dim1 [,dim2 [,dim3]])...]
```

Dimension String

```
DIM variable-name(length [,init-value])
[, variable-name (length [,init-value])...]
```

DIM - Dimension String Array

```
3.0 DIM array-name [elem1 [,elem2 [,elem3]]]
[(length[,init-value])...]
```

DIRECT - Define Single-Keyed-Access File

```
DIRECT file-name, key-size, num-records,
record-size, disk-num, sector-num
[,ERR=line-ref]
```

DISABLE - Prevent Logical Disk Access

```
DISABLE disk-num [,LOCAL] [,ERR=line-ref]
```

DROP - Drop Filename from File Table

```
DROP file-name [,ERR=line-ref]
```

DROP ALL - Drop All ADDRed Filenames from File Table

```
3.1 DROP ALL [ , ERR=line-ref ]
```

DUMP - Dump Data and Environment Information

```
1B2 DUMP keyword (channel [,ERR=line-ref]
dump-options)
```

Keywords:

ALL, HELP, ENVIRONMENT, ACTIVE PROGRAMS, STACKINFO, CALLSTACK, FORLOOPS, WHILELOOPS, GOSUBS, VARS, NUM VARS, STR VARS, INT VARS, ARRAYS, NUM ARRAYS, STR ARRAYS, INT ARRAYS, GLOBAL VARS, FORMATS, USER FNS, NUM USER FNS, STR USER FNS, RETRY, ESCAPE WHEN, ADDR PROGRAMS, WINDOWS, CHANNELS, FILES, PRINTERS, TERMINALS, IPLINFO, IPLFILE, IPLCONFIG, IPLPRMS, IPLDEVS, SYSTEM, SYSCOMMON

Dump Options:

"NAME=", "LEVEL="

EDIT - Program Editor

Full Screen Program Editor

EDIT

Program Line Editor

EDIT line-ref edit-specifier [string-constant]

Recall Last Line For Editing (Edit Recall)

8.3.1 ' [line-num]

EDITF - Formatted Program Editor

8.1 **EDITF**

ENABLE - Allow Logical Disk Access

ENABLE disk-ident [,LOCAL] [,ERR=line-ref]

ENCRYPT - Encrypt Program File

ENCRYPT program-name1, program-name2
[,PWD=passwd] [,ERR=line-ref]

END - End Program Execution

END

ENDTRACE - End Program Trace Mode

ENDTRACE

ENTER - Public Program Entry Point

ENTER [variable-list]

ERASE - Erase File

ERASE file-name [,ERR=line-ref]

ESCAPE - Escape Program Execution

ESCAPE

ESCAPE WHEN - Conditional Escape from Program Execution

8.2 **ESCAPE WHEN** condition

ESCOFF - Escape Trapping Off

8.1 **ESCOFF**

ESCON - Escape Trapping On

8.1 **ESCON**

EXECUTE - Execute Console Mode Instruction from Run Mode

EXECUTE string-value

EXIT - Exit from Public Program

EXIT [error-value]

EXITTO - Unconditional Branch & Clear Return Address

EXITTO line-ref

EXTRACT - Read Data & Lock Record

```
[P] EXTRACT (channel[,I/O-opts])[variable-  
list]  
[,IOL=line-ref]  
or  
[P]EXTRACT RECORD (channel[,I/O-opts])  
string-variable
```

I/O Options:

IND=numeric-value, KEY=string-value, SRT=sort-name,
ERR=line-ref, DOM=line-ref, END=line-ref, TBL=line-ref

FILE - Define a File from File Identification

```
FILE string-value
```

FIND - Read Data if Present

```
FIND (channel [,I/O-opts]) [variable-list]  
[,IOL=line-ref]  
or  
FIND RECORD (channel[,I/O-opts])string-  
variable
```

I/O Options:

IND=numeric-value, KEY=string-value, SRT=sort-name,
ERR=line-ref, DOM=line-ref, END=line-ref, TBL=line-ref

FINPUT

Field Input

```
3.0 FINPUT (channel, ATR=attribs  
[,EDT=edit-modes] [,TIM=seconds ]  
[,ERR=line-ref]) variable-name
```

Format for attribs:

byte 1?the attribute format
byte 2?starting column, unsigned binary
byte 3?starting row, unsigned binary
byte 4?length of one-row window, unsigned
binary
byte 5?length of data to accept from
window,
unsigned binary
byte 6?terminal attribute state for input
window
byte 7?foreground/background color
combination(See WINDOW COLOR
directive.)

<p>Format for edit-modes: byte 1?starting cursor position within field byte 2?termination mode off (\$00\$) or on (\$01\$) byte 3?clearing mode off (\$00\$) or on (\$01\$) byte 4?CTL value; unsigned binary. Refer to the CTL variable for values. byte 5?uppercase conversion off (\$00\$), for input characters only (\$01\$), or for input characters and existing field data (\$02\$)</p>

FIXUP - Fix Up Program File Structure

8.1B2 **FIXUP** program-name [,ERR=line-ref]

FLOATING POINT - Exponential Numeric Mode

FLOATING POINT

FOR/NEXT - Loop Controlled by Counter

FOR numeric-variable = numeric-value1
TO numeric-value2 [STEP numeric-value3]
NEXT numeric-variable

FORMAT DEFAULT - Initialize Format's Data Elements and Load Preset Values

8.2 **FORMAT DEFAULT** format-name
[,OPT = "DFONLY"] [,ERR=line-ref]
or
FORMAT DEFAULT ALL [,OPT = "DFONLY"]
[,ERR=line-ref]

FORMAT DELETE - Remove Format from Memory

8.2 **FORMAT DELETE** format-name [,ERR =line-ref]
or
FORMAT DELETE ALL [,ERR=line-ref]

FORMAT INCLUDE - Load Format into Memory from Data Dictionary

8.2 **FORMAT INCLUDE** format-name [,OPT=option-str]
[,ERR =line-ref]

FORMAT INIT - Initialize Data Elements of Included Format

8.2 **FORMAT INIT** format-name [,ERR=line-ref]
or
FORMAT INIT ALL [,ERR=line-ref]

GET - Read Disk by Sector

GET disk-num, sector-num, [,ERR=line-ref]
string-variable

GOSUB - Branch to Subroutine

GOSUB line-ref

GOTO - Unconditional Program Branch

GOTO line-ref

IF/THEN/ELSE/FI - Conditional Test

IF condition [THEN] stmt [ELSE stmt] [FI]

Refer to **Operators** specified in this Quick Reference.

INDEXED - Define Indexed Files

INDEXED file-name, num-records, record-size, disk-num, sector-num [,ERR=line-ref]

INITFILE - Initialize File

3.0 INITFILE file-name [,ERR=line-ref]

INPUT - Read Terminated by Enter

INPUT [EDT] [(channel [,I/O-opts])]
[@(column [,row])] [,mnemonic
[,mnemonic...]] [,output] [,variable-list
[:verification]]
[,IOL=line-ref]
or
INPUT [EDT] RECORD [(channel [,I/O-opts])]
string-variable

I/O Options:

ERR=line-ref, DOM=line-ref, END=line-ref, TBL=line-ref,
SIZ=numeric value, TIM=numeric-value, LEN=numeric-value,
IND=numeric-value, KEY=string-value
Refer to the **CTL Variable** for values.

INSERT ARRAY - Insert Elements of an Array

3.2 INSERT ARRAY array-name [(pos1,count1)
[, (pos2,count2)[, (pos3,count3)]]]

IOLISST - Input/Output Variable List

IOLIST [@(col [,row])] [,mnemonic
[,mnemonic...]] [,output] [,variable-list]
[,variable-list [:masking]] [,IOL=line-ref]

LET - Variable Assignment

[LET] variable-name = value

LET FMD - Store String Value into Format's Data Area

3.2 LET FMD (string-value [,element-number
[,occurrence-number]]) = data\$[,ERR=line-
ref]

LET FMT - Assign String Value to Data Name

```
LET FMT (str-val[,elem-num[,occ-  
num]])=data-val [,ERR=line-ref])
```

LIST - List Program Statements

```
LIST [(channel [,ERR=line-ref] [,IND=index-  
num] [,TBL=line-ref])] [line-ref1] [,line-  
ref2]
```

LOAD - Load Program to Memory

```
LOAD program-name [,PWD=password]
```

LOCK - Lock File for Exclusive Use

```
LOCK (channel [,ERR=line-ref])
```

LOG CLOSE - Close Transaction Log File

8.2 LOG CLOSE [,ERR=line-ref]

LOG OPEN - Open Log File for Entries

8.2 LOG OPEN filename,option[,ERR=line-ref]

Options: APPEND or REWIND

LONGVAR - Long Variable Name Entry Mode

8.0 LONGVAR

MERGE - Merge Program Lines from Separate Source

```
MERGE (channel [,ERR=line-ref] [,IND=index-  
num] [TBL=line-ref])
```

MSORT - Define Multi-Keyed File

8.1 MSORT file-name, [sort-name1:] sortdef1
[:mode1] [, [sort-name2:] sortdef2
[:mode2] [, ... [sort-name:n:] sortdefn
[:moden]]], num-records, record-size, disk-
num, sector-num [,ERR=line-ref]

ON GOSUB - Conditional Gosub

```
ON numeric-value GOSUB line-ref0 [,line-  
ref1 [,line-ref2...line-ref-n]]
```

ON GOTO - Conditional Goto

```
ON numeric-value GOTO line-ref0 [,line-ref1  
[,line-ref2...line-ref-n]]
```

OPEN - Open a File on I/O Channel

```
OPEN (channel [,ERR=line-ref]  
[,OPT=file-type] [,ISZ=record-size]  
[,SEP=field-sep] [,DEV=dev-string])file-  
name  
or  
OPEN (channel, OPT="DDE", ISZ=execute-mode)  
server|topic
```

PACK ARRAY - Pack String Array into String

8.3

PACK ARRAY array-name [ALL] ,str-var
[,pack-oper] [,ERR=line-ref]

PEXTRACT - Previous Extract

[P]**EXTRACT** (channel[,I/O-opts])[variable-
list] [,IOL=line-ref]
or
[P]**EXTRACT RECORD** (channel[,I/O-
opts])string-variable

I/O Options:

IND=numeric-value, KEY=string-value, SRT=sort-name,
ERR=line-ref, DOM=line-ref, END=line-ref, TBL=line-ref

PREAD - Previous Read

[P]**READ** (channel [,I/O-opts]) [variable-
list] [,IOL=line-ref]
or
[P] **READ RECORD** (channel [,I/O-opts]
string-variable

I/O Options:

IND=numeric-value, KEY=string-value, SRT=sort-name,
ERR=line-ref, DOM=line-ref, END=line-ref, TBL=line-ref

PRECISION - Set Numeric Precision

PRECISION numeric-value

PRINT - Output to Printer/Terminal

PRINT [(channel[,I/O-opts])] [@(column
[,row])] [,mnemonic [,mnemonic...]]
[,output] [,variable-list [:mask]]
[,IOL=line-ref]
or
PRINT RECORD [(channel[,I/O-opts])]string-
variable

I/O Options:

ERR=line-ref, DOM=line-ref, END=line-ref, TBL=line-ref,
IND=numeric-value, KEY=string-value

PROGRAM - Define Program File

PROGRAM file-name, program-size, disk-num,
sector-num [,ERR=line-ref]

PSAVE - Protected Save

[P] **SAVE** [program-name [,size, disk-num,
sector-num]] [,ERR=line-ref] [,PWD=passwd]

PUT - Write to Disk By Sector

PUT disk-num, sector-num [,ERR=line-ref],
string-variable [,verification]

READ - Read Data from I/O Channel

[**P**] **READ** [(channel [,I/O-opts])]
[variable-list] [,IOL=line-ref]
or
[**P**] **READ RECORD** [(channel [,I/O-opts])]
string-variable

I/O Options:

IND=numeric-value, KEY=string-value, SRT=sort-name,
ERR=line-ref, DOM=line-ref, END=line-ref, TBL=line-ref

RELEASE - Terminate Task Operation

RELEASE [task-id]
or

8.1B2 **RELEASE** [integer]

REM - Remarks

REM [comment]

REMOVE - Remove Record from File

REMOVE (channel, KEY=string-value [,I/O-
opts])

I/O Options:

DOM=line-ref, ERR=line-ref

REMSORT - Remove Sort Sequence

8.1 **REMSORT** file-name, SRT=sort-name[,ERR=line-
ref]

RENAME - Rename File

RENAME [disk-num,] old-file-name, new-file-
name [,ERR=line-ref]

RESERVE - Reserve Logical Disk for Exclusive Use

RESERVE disk-num [,ERR=line-ref]

RESET - Reset Program Environment

RESET

RETRY - Retry Statement that Caused Error Branch

RETRY

RETURN - Terminate Subroutine

RETURN

ROLLBACK - End Tracking Record Changes and Restore Records to Original State

3.2 **ROLLBACK** [,ERR=line-ref]

RUN - Run Program

RUN [program-name] [,ERR=line-ref]

SAVE - Save Program from Memory to Disk

[P] **SAVE** [program-name [,size, disk-num, sector-num]] [,ERR=line-ref] [,PWD=passwd]

SERIAL - Define Serial File

SERIAL file-name, num-records, record-size, disk-num, sector-num [,ERR=line-ref]

SET CMASK - Assigns Foreign Currency Parameters

3.2 **SET CMASK** currency-parms [,ERR=line-ref]

Currency Parameters: . =, \$ = parm
--

SET DATEMASK - Changes the System Date Format

3.2 **SET DATEMASK** string-value [,ERR = line-ref]

SET DATESTRINGS - Change SQL Month & Day Names

3.1 **SET DATESTRINGS** string-value [,ERR=line-ref]

SETDAY - Change System/Task Date

SETDAY string-value

SETDIR - Set Current Directory

3.1 **SET DIR** string-value [,ERR=line-ref]

SETDRIVE - Change Default DOS Disk

B2 **SETDRIVE** disk specifier [,ERR=line-ref]

SET ERC - Set Error Condition Variable

3.3 **SETERC** numeric-value

SETERR - Set Error Branch

SETERR line-ref|ON|OFF

SETESC - Set Escape Branch

SETESC line-ref

SET HOTKEY - Set Hotkey Value and Program to Execute

3.3 **SET HOTKEY** hotkey-value, "public-program"

SET PREFIX - Set Prefix Path Names

3.1 **SET PREFIX** string-value [,ERR=line-ref]

SET PRM - Sets Prm Options That are Flags

3.2 **SET PRM** string-value [,ERR=line-ref]

SETTIME - Change System/Task Time of Day

SETTIME numeric-value

SETTRACE - Start Program Trace Mode

SETTRACE [(channel)]

SET TRACEMODE - Sets The Mode of Tracing

8.2 **SET TRACEMODE** string [,ERR=line-ref]

SHORTVAR - Short Variablename Entry Mode

8.0 **SHORTVAR**

SORT - Define Sort File

SORT file-name, key-size, num-keys, disk-num, sector-num [,ERR=line-ref]

START - Initiate Task

START pages [,ERR=line-ref] [,BNK=bank-num] [,program-name] [,task-id]

STOP - Stop Program Execution

STOP

SYMTAB - Program Symbol Tables

8.1 **SYMTAB** program-specifier, string-array-name [ALL] [,ERR=line-ref]

Program-specifier: program-name, CURRENT, MAIN
--

SYSTEM - Temporary Exit to Operating System

SYSTEM [string-value]

TABLE - Data Conversion Table

TABLE mask table

TEXT - Define System Text File

8.1 **TEXT** file-name, disk-num, sector-num [,ERR=line-ref]

TISAM - Define Thoroughbred ISAM File

8.1 **TISAM** file-name, sortdef1
[:mode1][,sortdef2
[:mode2] [,...sortdefn [:moden]]],
num-records, record-size, disk-num, sector-
num
[, RR=line-ref]

TRANSACTION BEGIN - Begin Tracking Record Changes

8.2 **TRANSACTION BEGIN** [,ERR= line-ref]

UNLOCK - Unlock I/O Channel

UNLOCK (channel [,ERR=line-ref])

UNPACK ARRAY - Put Packed String into String Array

8.3 **UNPACK ARRAY** string-array,array-name[ALL] [,ERR=line-ref]

WAIT - Suspend Operation for Specified Time

WAIT seconds

WHILE - Loop Controlled by Conditional Test

3.1 **WHILE** condition **WEND**

Refer to **Operators** in this Quick Reference.

WINDOW ATTR - Set Window Attribute

3.1 **WINDOW ATTR** (attribute-num)

#	Attribute:			
0	background	normal video	underline off	blink off
1	foreground	normal video	underline off	blink off
2	background	reverse video	underline off	blink off
3	foreground	reverse video	underline off	blink off
4	background	normal video	underline on	blink off
5	foreground	normal video	underline on	blink off
6	background	reverse video	underline on	blink off
7	foreground	reverse video	underline on	blink off
8	background	normal video	underline off	blink on
9	foreground	normal video	underline off	blink on
10	background	reverse video	underline off	blink on
11	foreground	reverse video	underline off	blink on
12	background	normal video	underline on	blink on
13	foreground	normal video	underline on	blink on
14	background	reverse video	underline on	blink on
15	foreground	reverse video	underline on	blink on

WINDOW COLOR - Set Window Color

3.2 **WINDOW COLOR** (color-num)

BACKGROUND COLOR

<u>DEC</u>	<u>HEX</u>	<u>COLOR</u>	<u>DEC</u>	<u>HEX</u>	<u>COLOR</u>
000	\$00\$	BLACK	016	\$10\$	LT. BLUE
032	\$20\$	LT. GREEN	048	\$30\$	LT. CYAN
064	\$40\$	LT. RED	080	\$50\$	LT. MAGENTA
096	\$60\$	YELLOW	112	\$70\$	LT. GRAY
128	\$80\$	GRAY	144	\$90\$	BLUE
160	\$A0\$	GREEN	176	\$B0\$	CYAN
192	\$C0\$	RED	208	\$D0\$	MAGENTA
224	\$E0\$	BROWN	240	\$F0\$	WHITE

FOREGROUND COLOR

<u>DEC</u>	<u>HEX</u>	<u>COLOR</u>	<u>DEC</u>	<u>HEX</u>	<u>COLOR</u>
00	\$00\$	BLACK	01	\$01\$	LT. BLUE

02	\$02\$	LT. GREEN	03	\$03\$	LT. CYAN
04	\$04\$	LT. RED	05	\$05\$	LT. MAGENTA
06	\$06\$	YELLOW	07	\$07\$	LT. GRAY
08	\$08\$	GRAY	09	\$09\$	BLUE
0	\$0A\$	GREEN	11	\$0B\$	CYAN
12	\$0C\$	RED	13	\$0D\$	MAGENTA
14	\$0E\$	BROWN	15	\$0F\$	WHITE

WINDOW CREATE - Create New Window

- 8.1 **WINDOW CREATE** (width, height, coll, row1)
[attributes]

Keywords:

"BORDER=", "BORDERATR=", "TITLE=", "TITLEAT=",
"NAME=", "INIT=", "INITATR=", "WRAP=", "SCROLL=",
"SELECTATR=", "PAINTMODE=", "PANELCOUNT=",
"CONTENTS=", "INITCOLOR=", "BORDERCOLOR=",
"LICOLOR="

WINDOW DELETE - Delete Window

- 8.1 **WINDOW DELETE** (TBWin-name)

WINDOW FKEYS - Reload Function Keys

- 8.2 **WINDOW FKEYS** (fkey-values) ["NAME=TBWin-name"]

Format of fkey-values:

byte meaning

1 function key number

2 string length

Remaining bytes make up the new function key string.

WINDOW GETINFO - Collect Window Manager Status

- 8.1 **WINDOW GETINFO** (array-name [ALL])
["NAME=TBWin-name"]

WINDOW IOREGION - Create Protected Region within Window

- 8.1 **WINDOW IOREGION** (CREATE, width, height,
coll, row1)

WINDOW IOREGION - Delete Protected Region within Window

- 8.1 **WINDOW IOREGION** (DELETE)

WINDOW MOVE - Move Window on Screen

- 8.1 **WINDOW MOVE** (coll, row1) ["NAME=TBWin-name"]

WINDOW PANEL - Working with a Protected Panel within Window

Define Protected Panel within Window

- 8.2

WINDOW PANEL (CREATE, width, height, col1, row1, panel-name) [attributes]

Keywords:

"BORDER=", "BORDERATR=", "TITLE", "TITLEAT=", "INIT=", "INITATR=", "WRAP=", "SCROLL=", "INITCOLOR=", "BORDERCOLOR="

Delete Protected Panel within Window

3.2 **WINDOW PANEL** (DELETE, panel-name)

Select Protected Panel within Window

3.2 **WINDOW PANEL** (SELECT, panel-name)

Turn off Protected Panel within Window

3.2 **WINDOW PANEL** (OFF)

WINDOW POP

Delete Current Window

3.1 **WINDOW POP**

WINDOW PUSH

Create Duplicate of Current Window

3.1 **WINDOW PUSH** [?NAME=TBWin-name?]

WINDOW PUT - Reprint Current Window

3.1 **WINDOW PUT** [delim-1] (map-string)

3.1 **WINDOW PUT** [delim-1] delim-2

WINDOW REFRESH - Reprint Entire Screen

3.1 **WINDOW REFRESH**

WINDOW RESIZE - Change Window Size

3.2 **WINDOW RESIZE** (width, height) [TBWin-name] [,] [up-down, left-right]

WINDOW RESTORE - Reprint & Activate Window

3.2 **WINDOW RESTORE** (TBWin-name)

WINDOW SAVE - Save Window

3.2 **WINDOW SAVE** ([POP,] TBWin-name)

WINDOW SCROLL - Move Data within Window by Row/Column

3.1 **WINDOW SCROLL** (ON)

3.1 **WINDOW SCROLL** (OFF)

3.1 **WINDOW SCROLL** (LEFT, col1)

3.1 **WINDOW SCROLL** (RIGHT, col1)

3.1 **WINDOW SCROLL** (UP, row1)

3.1 **WINDOW SCROLL** (DOWN, row1)

WINDOW SELECT - Activate Window

8.1 **WINDOW SELECT** ([NOUPDATE,] TBWin-name)

WINDOW SHAPE - Draw Shape within Window

8.1 **WINDOW SHAPE** (BOX, width, height, coll, rowl) [attributes]

Keywords:

"BORDER=", "BORDERATR=", "TITLE", "TITLEAT=", "INIT=", "INITATR=", "INITCOLOR=", "BORDERCOLOR="

8.1 **WINDOW SHAPE** (LINE, direction, coll, rowl, length)

WINDOW SWAP - Exchange Active Status with Last Window

8.1 **WINDOW SWAP**

WINDOW WRAP - Change Wrap Attribute

8.1 **WINDOW WRAP** (ON)

8.1 **WINDOW WRAP** (OFF)

WRITE - Write Data to I/O Channel

WRITE [(channel [,I/O-opts][,OPT="LOCK"])]
[,variable-list [:mask]] [,IOL=line-ref]
or
WRITE RECORD [(channel [,I/O-opts]
[,OPT="LOCK"])] string-variable

I/O Options

ERR= line-ref, DOM=line-ref, END=line-ref, TBL=line-ref,
IND=numeric value, KEY= string-value

XCALL - Call External Programs

8.3 **XCALL** function-name [,ERR=line-ref]
[,format-name, arg1[, arg2]...]

Functions & System Variables

Numeric Function String Function System Variable

ABS - Absolute Value (Numeric Function)

ABS (numeric-value [,ERR=line-ref])

ACS - Arc Cosine (Numeric Function)

ACS (numeric-value [,ERR=line-ref])

8.0 **=ALL - Equal Repeated Character (String Function)**

=ALL (string-value)

AND - Logical AND (String Function)

AND (string-value1,string-value2
[,ERR=line-ref])

8.1B2 **ARG - Arguments for Basic Startup (String Function)**

ARG (numeric value [,ERR=line-ref])

B2 **ARGC - Argument Count for *Basic* Startup (System Variable)**

ASC - ASCII (Numeric Function)

ASC (string-value [,ERR=line-ref])

ASN - Arc Sine (Numeric Function)

ASN (numeric-value [,ERR=line-ref])

ATH - ASCII to Hexadecimal (String Function)

ATH (string-value [,ERR=line-ref])

ATN - Arc Tangent (Numeric Function)

ATN (numeric-value [,ERR=line-ref])

ATQ - Arc Tangent of a Quotient (Numeric Function)

ATQ (numeric-value1, numeric-value2
[,ERR=line-ref])

8.2 **ATR - Returns Data Element's Attribute Value (System Variable)**

ATR (name\$, elem-number, attr-number
[,ERR=line-ref])

<u>Attr-#</u>	<u>Attr-value</u>	<u>Attr-#</u>	<u>Attr-value</u>
0	data element value	15	pre-process proced.
1	length	16	Y/N value
2	precision	17	key indicator
3	numeric type	18	occurrence value
4	field separator	19	documentation code
5	type input	20	position in format
6	pad	21	data element name
7	date	22	occurrence field sep.
8	audit	23	print length-no comma mask
9	post-process proced.	24	print length-comma mask
10	special prompt	25	mask with no commas
11	preset value	26	mask with commas
12	valid value	27	data description
13	delete value	28	language code
14	security value		

BIN - Binary (String Function)

BIN (numeric-value,result-length
[,ERR=line-ref])

BSZ - Bank Size (Numeric Function)

BSZ (bank-num [,ERR=line-ref])

3.0 **CDN - Current Date Numeric (System Variable)**

3.0 **CDS - Current Date String (System Variable)**

8.1 **CGV - Common Global Variable (String Function)**

CGV (string-value-1 [,string-value-2] [,ERR=line-ref])

Operations:
"!CLEARALL"? delete unprotected global variables
"!CLEAR"? delete a protected or unprotected global variable name specified in string-value2
"!CLEARTO"? delete unprotected global variable specified in string-value2 and all other unprotected global variables created or changed since the last operation on the specified global variable.

CHR - Character (String Function)

CHR (numeric-value [,ERR=line-ref])

8.2 **CMASK - Returns Foreign Currency Parameters (System Variable)**

COS - Cosine (Numeric Function)

COS (numeric-value [,ERR=line-ref])

CPL - Convert Thoroughbred *Basic* Statement to Compiled Format (String Function)

CPL (string-value [,ERR=line-ref])

8.1B2 **CPP - Compile Program (String Function)**

CPP (program-string [,ERR=line-ref])

Format of program-string:
1-8 name of program
9-10 length of first BASIC program line; unsigned binary
11-n first BASIC program line in listed format
? length and listed format repeat as necessary?

CRC - Cyclic Redundancy Code (String Function)

CRC (string-value [,2-byte-string] [,ERR=line-ref])

CTL - Control Key Variable (System Variable)

CTL VARIABLE VALUES:

<u>Key</u>	<u>Value</u>	<u>Key</u>	<u>Value</u>
<CR> <Enter>	0	Character insert	? 7
<F1>	1	Line insert	? 8
<F2>	2	Line delete	? 9
<F3>	3	Line erase	? 10
<F4>	4	Page erase	? 11
<F5>	5	Tab	? 12
...	...	Back tab	? 13
<Fn>	n	Home	? 14
Right arrow	? 1	< Control > P	? 15
Left arrow	? 2	Page down	? 16
Down arrow	? 3	Page up	? 17

Up arrow	? 4	Screen down	? 18
Back space	? 5	Screen up	? 19
Character delete	? 6	Boundary down	? 20
		Boundary up	? 21

Note: Keys with negative CTL values are returned only when the INPUT directive is used with the EDT option.

CVT - Convert String (String Function)

CVT (string-value, option-value
[,ERR=line-ref])

Option Operation

0	Do NOT edit.
1	Clear high-order bit in each byte.
2	Remove all blanks and tabs.
4	Remove unprintable characters below space (\$00\$ through \$1F\$).
8	Remove leading spaces & tabs.
16	Reduce each multiple occurrence of spaces & tabs to 1.
32	Convert lowercase characters to uppercase.
64	Convert ?[? & ?]? to ?(? & ?)?.
128	Remove trailing spaces & tabs.
256	Do not alter characters within double quotes.
512	Swap bytes in every 2-byte pair.
1024	Remove characters that are not spaces, alphabetic or numerics.
2048	Same as option-value=1, but do not alter the field separator for data records.
4096	Convert uppercase characters to lowercase.
8192	Convert the entire string into its mirror image.

3.2 DATEMASK - Returns SQL DATEMASK (System Variable)

3.1 DATESTRINGS - SQL MONTH & DAY Names (System Variable)

DAY - System/Task Date (System Variable)

3.1 DCM - Data Compress (String Function)

DCM (string-expression [,ERR=line-ref])

DEC - Decimal (Numeric Function)

DEC (string-value [,ERR=line-ref])

3.0 DIM - Dimension Function (String Function)

DIM (length [,value] [,ERR=line-ref])

3.1 DIR - Current Directory Variable (System Variable)

3.2 DNE - Data Name in Error (System Variable)

DSD - Device Status Description (String Function)

DSD (string-value [,ERR=line-ref])

8.1B2 **DSK - Current/Configured Disk Drives (String Function)**

`DSK (disk-specifier [,ERR=line-ref])`

DSZ - Sata Size - Available User Memory (System Variable)

8.0 **DTN - Date/Time Numeric (Numeric Function)**

`DTN (string-value [,date-mask] [,ERR=line-ref])`

Refer to **Date Masking Characters** in this Quick Reference.

8.1 **DTR - Data to Record Conversion (String Function)**

`DTR (expanded-record, data-defn-table [,ERR=line-ref] [,SEP=field-sep])`

EPT - Base 10 Exponent (Numeric Function)

`EPT (numeric-value [,ERR=line-ref])`

8.3 **ERC - Error Condition Variable (System Variable)**

8.1 **ERM - Error Message (String Function)**

`ERM (numeric-value [,ERR=line-ref])`

ERR - ERR Function (Numeric Function)

`ERR (error-list)`

ERR - ERR Variable (System Variable)

8.2 **ERRBUF - Error Buffer (System Variable)**

ESC - Escape Character (System Variable)

EXP - Natural Logarithm Exponent (Numeric Function)

`EXP (numeric-value [,ERR=line-ref])`

8.1 **FDTV - Open File Table Entries (System Variable)**

FID - File Identification (String Function)

`FID (channel [,ERR=line-ref])`

For files? format for 22-byte string:

<u>Byte(s)</u>	<u>Description</u>
1-3	If Btrieve file, byte 1="B". Otherwise unused (\$000000\$)
4-9	File name, first 6 characters
10	File type:\$00\$? Indexed file; \$01\$? Serial file; \$02\$? Direct or Sort file; \$03\$? TEXT file; \$04\$? Program file; \$06\$? MSORT file; \$07\$? TISAM & compatible; \$0A\$? Logical Disk Directory; \$0B\$? System file
11	Binary sum of Key Size <\$IKey;Size>+ Pointer Size:\$00\$ for Serial, Index, and

Program files; For Sort and Direct files:
 Key Size + 4 for less than 32,768 records;
 Key Size + 6 for more than 32,767 records

12-14 Binary number of records<\$INumber;of
 Records><\$IRecords, Number of> in file;
 Program and System files=\$000001\$

15-16 Binary number of bytes per Record;Sort=
 \$0000\$; Program=always a multiple of 256

17-19 Binary number of 256-byte sectors

20 Binary Logical Disk Directory number

21-22 File name, last 2 characters

23+ See bytes 23+ for logical disk directories

For logical disk directories? Format for 22- or 46-
 byte string:

<u>Byte(s)</u>	<u>Description</u>
1-10	Same as files
11	Unused (\$00\$)
12-14	Same as files
15-16	Binary number of bytes per Record
17-19	Unused (\$000000\$)
20-22	Same as files
23+	Additional bytes of directory path name

For tasks and devices? format for 2-byte string:

<u>Byte(s)</u>	<u>Description</u>
1-2	Task or device name

FIX - Special Integer Function (Numeric Function)

FIX (numeric-value [,ERR=line-ref])

FKY - First Key (System Variable)

FKY (channel [,SRT=sort-name] [,END=line-
 ref] [,ERR=line-ref])

3.2 FMD - Retrieves a Portion of the Data Area of a Format (String Function)

FMD (string-value [[,element-number]
 [,occurrence-number]] [,ERR=line-ref])

3.3 FMT - Retrieves Data Name from String (String Function)

FMT (string-value [[,element-number]
 [,occurrence-number]][,ERR=line-ref])
 [:format-mask]

8.2 FMTNL - Format Name List (System Variable)

FNx - Invoke User-Defined Function (Numeric Function)

FNx (value-list)

FNx\$ - Invoke User Defined Function (String Function)

(value-list)

FPT - Fractional Portioin (Numeric Function)

FPT (numeric-value [,ERR=line-ref])

8.1B2 FST - File System Information (String Function)

FST (full-path-name, option[,ERR=line-ref])

Format of Information returned:

<u>Byte(s)</u>	<u>Description</u>
1-2	System type: \$0020\$ = DOS; \$0021\$ = VAX; \$0022\$ = UNIX
3-4	Specific system type (not yet formulated; currently \$0000\$)
5-6	Thoroughbred <i>Basic</i> file type (not returned by option 0): \$0000\$ = INDEXED; \$0001\$ = SERIAL; \$0002\$ = DIRECT; \$0003\$ = TEXT; \$0004\$ = PROGRAM; \$0006\$ = MSORT; \$0007\$ = TISAM; \$000A\$ = Directory; \$000B\$ = System
7-10	Reserved for UNIX operating systems
11-14	Device inode resides on
15-18	This inode number
19-20	File protection bits
21-22	Host system file type: \$000A\$ = Directory File; \$000B\$ = System File; \$0040\$ = UNIX Block Special; \$0041\$ = UNIX Character Special; \$0042\$ = UNIX FIFO; \$0043\$ = UNIX Network Special; \$0044\$ = UNIX Socket; \$0045\$ = UNIX Symbolic Link (bytes 119 + point to the symbolic link name); \$004F\$ = Unknown device (to Thoroughbred <i>Basic</i>)
23-24	Number of links to file
25-26	User ID of owner
27-28	Group ID of owner
29-32	Device ID(valid only for Block and character special devices)

33-36	File size in bytes
37-42	File Status change date/time; SQL format
43-48	Last access date/time; SQL format
49-54	Last modified date; SQL format
55-118	Reserved

GAP - Generate Odd Parity (String Function)

GAP (string-value[,ERR=line-ref])

HSH - Hash (String Function)

HSH (string-value [,2-byte-string]
[,ERR=line-ref])

HTA - Hexadecimal to ASCII (String Function)

(string-value [,ERR=line-ref])

IND - Index (Numeric Function)

IND (channel[,ERR=line-ref, END=line-ref])

B2 INF - System and Task Information (String Function)

INF (numeric-value1, numeric-value2
[,ERR=line-ref])

<u>Numeric-Value 1,2</u>	<u>Information Returned</u>
0,0	Operating system name
0,1	Operating system level
1,0	CPU id
3,0	UNIX: 2-byte unique process id of this Thoroughbred Basic task; unsigned binary
3,1	UNIX: same as 3,0 except returned as the first 2 bytes of an 8-byte string; last 6 bytes are null (\$00\$)
3,2	User's login id
3,3	User's name
4,n	User's environment; n is an integer in the range of 0 to the maximum number of environment strings minus 1.

INT - Integer (Numeric Function)

INT (numeric-value [, ERR=line-ref])

IOR - Binary Inclusive OR (String Function)

IOR (string-value1, string-value2
[,ERR=line-ref])

KEY - Key Value (String Function)

KEY (channel [,SRT=sort-name] [,END=line-
ref] [,ERR=line-ref])

LEN - String Length (Numeric Function)

LEN (string-value [,ERR=line-ref])

LKY - Last Key (String Function)

LKY (channel [,SRT=sort-name] [,END=line-ref] [,ERR=line-ref])

LOG - Logarithm (Numeric Function)

LOG (numeric-value)

LRC - Longitudinal Redundancy Check (System Variable)

LRC (string-value [,ERR=line-ref])

LST - Convert Thoroughbred *Basic* Statement to List Format (System)

LST (string-value [,ERR=line-ref])

8.1 MAX - Maximum Numeric Value (Numeric Function)

MAX (numeric-value1 [,numeric-value2 [,...numeric-valuen]])

8.1 MIN - Minimum Numeric Value (Numeric Function)

MIN (numeric-value1 [,numeric-value2 [,...numeric-valuen]])

8.0 MNE - Mnemonic (System Variable)

MNE (mnemonic-code [,channel [,ERR=line-ref])

MOD - Modulus - Division Remainder (Numeric Function)

MOD (numeric-dividend, numeric-divisor [,ERR=line-ref])

8.1 NEA - String/Numeric Array Data (Numeric Function)

NEA (array-name, numeric-code [,ERR=line-ref])

<u>Numeric-Code</u>	<u>Value Returned</u>
-3	lower bound of the 3rd dimension
-2	lower bound of the 2nd dimension
-1	lower bound of the 1st dimension
0	number of dimensions in the array
1	number of elements in the 1st dimension
2	number of elements in the 2nd dimension
3	number of elements in the 3rd dimension

NLG - Natural Logarithm (Numeric Function)

NLG (numeric-value [,ERR=line-ref])

8.1 NMV - Numeric Value (Numeric Function)

NMV (string-value)

NOT - Binary Inversion (String Function)

NOT (string-value [,ERR=line-ref])

3.0 **NTD - SQL Number to String Date (String Function)**

NTD (numeric-value[,date-mask][,ERR=line-ref])

Refer to **Date Masking Characters** in this Quick Reference.

NUM - Numeric (Numeric Function)

NUM (string-value, NTP = numeric-type
[,ERR=line-ref])

or

NUM (string-value[,ERR=line-ref])

<u>NTP</u>	<u>Description</u>
0	Fixed point positive/negative numbers
1	Fixed point positive numbers
2	Fixed point negative numbers
3	Binary positive/negative numbers
4	Binary positive numbers
5	Binary negative numbers
6	Packed decimal numbers
7	Informix decimal numbers
8	IEEE single precision floating point
9	IEEE double precision floating point
10	BCD signed
11	BCD unsigned
12	BCD no sign byte
13	ASCII sign stored in the high nibble of last byte
14	ASCII sign leading separate
15	ASCII sign trailing separate

3.0 **OCH - Open Channel (System Variable)**

3.1 **PAD - Justify & Pad String (String Function)**

PAD (string-value, numeric-value [,pad-option] [,pad-value] [,ERR=line-ref])

PCK - Pack Integer into String (String Function)

PCK (numeric-value, length [,ERR=line-ref])

3.0 **PFL - Prepare for Listing (String Function)**

PFL (string-value, symbol-table)

3.0 **PFP - Prepare for Program (String Function)**

PFP

PFP (string-value, symbol-table)

B2 **PGCHARBASE - Portable Graphics Character Base (System Variable)**

PGM - Return Thoroughbred *Basic* Statement in Compiled Format (String Function)

PGM (numeric-value [,MAIN])

PGN - Program Name (System Variable)

PKY - Previous Key (String Function)

PKY (channel [,SRT=sort-name] [,END=line-ref] [,ERR=line-ref])

POS - Scan String & Return Position (Numeric Function)

POS (search-string relational-operator reference-string [,step-value [,occurrence]])

8.0 PRC - Precision Variable (System Variable)

8.1 PREFIX - Prefix Path Names (System Variable)

8.2 PRM - Returns PRM Options That Are Flags (System Variable)

Format of string value returned:

<u>Byte</u>	<u>Bit</u>	<u>PRM flag</u>
1	\$80\$	UPPER
1	\$40\$	DISABLE
1	\$20\$	ALLOC
1	\$10\$	ERRMASK
1	\$08\$	FULL-COMPARE
1	\$04\$	IF47
1	\$02\$	LONG-PROMPT
1	\$01\$	NOROUND
2	\$80\$	READONLY
2	\$40\$	OFF-ERR127
2	\$20\$	SERIAL-EOF
2	\$10\$	LISTPAREN
2	\$08\$	DONTCHECKTEXT
2	\$04\$	SLEEPLOCK
2	\$02\$	SHORT-ERROR
2	\$01\$	VAR-NOTSET-ERR
3	\$80\$	NOTRANS
3	\$40\$	IEEE SWAP
4	ALL	RESERVED

PSZ - Program Size (System Variable)

8.1 PTN - Memory Partition Size (System Variable)

PUB - Public Program (String Function)

PUB (bank-num [,ERR=line-ref])

QUO - Quote (System variable)

RND - Random (Numeric Function)

RND (numeric-value [,ERR=line-ref])

8.1 RTD - Record to Data Conversion (String Function)

RTD (data-record, data-defn-table
[,ERR=line-ref] [,SEP=field-sep])

B2 **SDX - Soundex Value of String (String Function)**

SDX (string-value [,ERR=line-ref])

SEP - Field Separator Character (System Variable)

SGN - Determine Sign of Numeric Value

SGN (numeric-value [,ERR=line-ref])

SIN - Sine of an Angle (Numeric Function)

SIN (numeric-value [,ERR=line-ref])

SQR - Square Root

SQR (numeric-value [,ERR=line-ref])

SSN - Software Serial Number

SSZ - Sector Size (Numeric Function)

SSZ (disk-num [,ERR=line-ref])

3.1 **STL - String Length (Numeric Function)**

STL (string-variable)

STR - Convert Numeric to String Value

STR (numeric value[:format-mask]
[,ERR=line-ref])

or

STR (numeric-value, NTP=numeric-type,
SIZ=number-bytes[,ERR=line-ref])

NTP Description and Maximum Storage Size

0	Fixed point positive/negative numbers: 16
1	Fixed point positive numbers: 16
2	Fixed point negative numbers: 16
3	Binary positive/negative numbers: 8
4	Binary positive numbers: 8
5	Binary negative numbers: 8
6	Packed decimal numbers: 6
7	Informix decimal numbers: 8
8	IEEE single precision floating point: 4(only)
9	IEEE double precision floating point: 8(only)
10	BCD signed: 8
11	BCD unsigned: 8
12	BCD no sign byte: 7
13	ASCII sign stored in the high nibble of last byte: 14
14	ASCII sign leading separate: 15
15	ASCII sign trailing separate: 15

3.1 **SWP - Byte Swap (String Function)**

SWP (string-value, swap-option [,ERR=line-ref])

Swap-options:

<u>Bit</u>	<u>Function</u>
0	swap adjacent bytes
1	swap adjacent 2-byte words
2	swap adjacent 4-byte long words.

SYS - System Name (System Variable)

TAN - Tangent (Numeric Function)

TAN (numeric-value [,ERR=line-ref])

8.1 TBL - Table Function (String Function)

TBL (string-value, table-string[,ERR=line-ref])

TBL (string-value, TBL=line-ref[,ERR=line-ref])

TCB - Task Control Block (Numeric Function)

(numeric-value [,ERR=line-ref])

<u>Numeric Value</u>	<u>Value Returned</u>
n	where n = 0,1,2, are undefined and normally returns 0
3	returns the last operating system error code
4	returns the current statement number
5	returns the # of the last statement that caused an error
6	returns the statement that SETESC is set to
7	returns the statement that SETERR is set to
8	returns the shell exit code from the last SYSTEM directive
9	returns the shell termination status
13	returns the current public program level. The value is the actual number of public programs in the stack. While in the main program this value is 0.

TIM - Task Time in Hours & Decimal Hours (System Variable)

8.2 TRACEMODE - Sets the Mode of Tracing (System Variable)

TSK - Bank Task Data (String Function)

TSK (bank-num [,ERR=line-ref])

TSK - System Task Data

TSK (0 [,ERR=line-ref])

TSK - Report Active Ghost Task(S)

TSK (2 [,ERR=line-ref])

TSM - Termination Status Message (System Variable)

3.1 UCM - Uncompress Data (String Function)

```
UCM (string-expression [,ERR=line-ref])
```

3.3 UNT - Lowest Channel Number for File (Numeric Variable)

```
UNT (file-name)
```

3.0 UNT - Unused Channel (System Variable)

UPK - UNPACK Integer from String (String Function)

```
UPK (string-value [,ERR=line-ref])
```

3.1 WIN (GET) - Collect Thoroughbred *Basic* Window Data/Attributes (String Function)

```
WIN (GET [delim-1] [delim-2])
```

3.1 WIN (GETCURSOR) - Collect Thoroughbred Window Cursor Position (String Function)

```
WIN (GETCURSOR [,PHYSICAL])
```

3.1 WIN (GETLIST) - Collect List of Active Thoroughbred *Basic* Windows (String Function)

```
WIN (GETLIST)
```

3.2 WIN (GETSAVEDLIST) - Collect List of Saved Thoroughbred *Basic* Windows (String Function)

3.1 WIN (GETSCREEN) - Collect Entire Screen Attributes

3.1 XFD - Extended File Identification (String Function)

```
XFD (channel, option [,ERR=line-ref])
```

Options:

#	data returned
0	general data for file or directory
1	general data for device
3	key definition data for MSORT and TISAM files
5	general data about the file to be used with a FILE or INITFILE directive
6	alternate-format FID data
10	the number of records in use

XOR - Binary Exclusive OR (String Function)

```
XOR (string-value1,string-value2  
[,ERR=line-ref])
```

Dictionary-IV APIs

Use the appropriate command syntax below to execute an API.

READ or WRITE Text Objects

```
CALL "OO41", TEXT$[ALL], LNK$[ALL],  
LNK[ALL]
```

On-line Calculator

```
CALL "8CALC", VALUE, VALUE$, ]SYSV$
```

Closes a Screen, View, or Format

```
CALL "8CLOSE", FUNC$, SCREEN$[ALL],  
FORMAT$[ALL], ]SYSV$
```

Selects Colors for Thoroughbred *Basic* Windows

```
CALL "8COLORP", FUNC$, ATTR, COLR, ]SYSV$
```

Creates, Erases, or Renames Files

```
CALL "8FILEA", FUNC$, PARM$, ]SYSV$
```

Expands, Modifies, Copies, or Moves Files

```
CALL "8FILEB", FUNC$, ]SYSV$
```

Reads, Adds, or Updates *Dictionary-IV* Formats

```
CALL "8FORMAT", MSG$[ALL], RV$, ]SYSV$
```

Prints Help Window

```
CALL "8HELP", FUNC$, HELP$, HELPTXT$,  
]SYSV$
```

Inputs Screen Data

```
CALL "8INPUT", FUNC$, SCREEN$[ALL],  
FORMAT$[ALL], DATA$, ]SYSV$
```

Prints and Selects Options from a Menu

```
CALL "8MENU", FUNC$, MENU$, MENUSEL$,  
]SYSV$,  
PPA$[ALL]
```

Moves a window on screen

```
CALL "8MOVE", MPARMS$, ]SYSV$
```

Processes Operator Messages

```
CALL "8MSG", FUNC$, MT$[ALL], C, ]SYSV$
```

Opens Printer

```
CALL "8OPENP", FUNC$, PT$[ALL], PCH,  
]SYSV$
```

Opens Screen, View, or Format

```
CALL "8OPENS", FUNC$,  
SCREEN$[ALL], FORMAT$[ALL], DATA$, ]SYSV$
```

Prints Screen or Screen Data

```
CALL "8PRINT", FUNC$, SCREEN$[ALL],  
FORMAT$[ALL], DATA$, ]SYSV$
```

Resizes Thoroughbred *Basic* Windows

```
CALL "8RSIZE", MPARMS$, ]SYSV$
```

Maintains Text Records from a Data File

```
CALL "8TEXTF", FUNC$, SCREEN$[ALL],  
FORMAT$[ALL], C, ]SYSV$
```

Reads Text Records from a Data File

```
CALL "8TEXTR", FUNC$, TPARMS$[ALL],  
TEXT$[ALL], ]SYSV$
```

Prints and Selects Records from a Data File

```
CALL "8VIEWF", FUNC$, SCREEN$, [ALL],  
FORMAT$, [ALL], C, ]SYSV$
```

Prints and Selects Options from an Internal Table

```
CALL "8VIEWT", FUNC$, VIEWT$, [ALL], ]SYSV$
```

Prints Hard Copy of Screen

```
CALL "8ZPHC", FUNC$, TH$, TEXT$, [ALL],  
HDNG$, [ALL], PT$, [ALL], PC, ]SYSV$
```

I/O MNEMONICS

Mnemonics		Window	Description
Start	End	Only	
A1		✓	Attribute Lead-in Sequence
A2		✓	Attribute Number of Bytes
A3		✓	Attribute Values
A4		✓	Attribute Trailing Sequence
A5		✓	Number of Attribute States
A6		✓	Auto-wrap Terminal
A7		✓	Reserved
A8		✓	Attribute Space Indicator
A9		✓	Color algorithm for the terminal
AA		✓	Attribute Lead-in Sequence (ANSI)
AB		✓	Attribute Values (ANSI)
AC		✓	Attribute Separator (ANSI)
Mnemonics		Window	Description
Start	End	Only	
AD		✓	Attribute Trailing Sequence (ANSI)
AE		✓	Number of Attribute States (ANSI)
AF		✓	Attributes Off (ANSI)
BACKGR		✓	Changes Background Color
BB	EB		Blink
BD	EB		Blink Underline
BE	EE	✓	Echo
BF	ER		Reverse Foreground
BG	EG		Graphics Mode
BI	EI	✓	Input Transparency
BLACK		✓	Black Foreground Color
BLUE		✓	Blue Foreground Color
BM	EM	✓	Error 29 (Undefined Mnemonic)
BO	EO	✓	Output Transparency
BR	ER	✓	Reverse Background
BROWN		✓	Brown Foreground Color
BS			Backspace
BT	ET	✓	Type-ahead
BU	EU		Underline

BV	EB		Blink Reverse
CE			Clear to End of Window
CF			Clear Foreground
CH			Cursor Home
CI		✓	Clear Input Buffer
CL			Clear to End of Line
CLI		✓	Color associated with low intensity
CN	CO		Cursor On/Off
CR			Carriage Return
CS			Clear Screen
CU		✓	Read Cursor
CYAN		✓	Cyan Foreground Color
DC		✓	Delete Character
DM		✓	Set Window Defaults
DN	DO		Screen On/Off
EF		✓	End with Foreground intensity
EP			Begin Expanded Print Mode
ES			Escape Character
EU		✓	End Underline mode
EX		✓	End with either Foreground or Background intensity
FF			Printer Form Feed Character
G0			Horizontal Line
G1			Vertical Line
G2			Upper Left Corner
G3			Upper Right Corner
G4			Lower Left Corner
G5			Lower Right Corner
G6			Connect to Left
Mnemonics		Window	
Start	End	Only	Description
G7			Connect to Right
G8			Connect to Lower
G9			Connect to Upper
GA			Cross in Middle
GB			High Intensity Block
GC			Mid Intensity Block
GD			Low Intensity Block
GE			Double Vertical Bar
GF			Double Horizontal Bar
GRAY		✓	Gray Foreground Color
GREEN		✓	Green Foreground Color
IC		✓	Insert Character
K0		✓	Middle Part Character Sequence for Black
K1		✓	Middle Part Character Sequence for Light Blue
K2		✓	Middle Part Character Sequence for Light

			Green
K3		✓	Middle Part Character Sequence for Light Cyan
K4		✓	Middle Part Character Sequence for Light Red
K5		✓	Middle Part Character Sequence for Light Magenta
K6		✓	Middle Part Character Sequence for Yellow
K7		✓	Middle Part Character Sequence for Light Gray
K8		✓	Middle Part Character Sequence for Gray
K9		✓	Middle Part Character Sequence for Blue
KA		✓	Middle Part Character Sequence for Green
KB		✓	Middle Part Character Sequence for Cyan
KC		✓	Middle Part Character Sequence for Red
KD		✓	Middle Part Character Sequence for Magenta
KE		✓	Middle Part Character Sequence for Brown
KF		✓	Middle Part Character Sequence for White
KW		✓	Start Foreground Color
KX		✓	End Foreground Color
KY		✓	Start Background Color
KZ		✓	End Background color
LBLUE		✓	Light Blue Foreground Color
LC	UC	✓	Lower/Upper Case
LCYAN		✓	Light Cyan Foreground Color
LD			Line Delete
LF			Line Feed
LGRAY		✓	Light Gray Foreground Color
LGREEN		✓	Light Green Foreground Color
Mnemonics		Window	
Start	End	Only	Description
LI			Line Insert
LMAGENTA		✓	<%-3>Light Magenta Foreground Color
LRED		✓	Light Red Foreground Color
MAGENTA		✓	Magenta Foreground Color
MN	MO	✓	Window Mgr. Mapping On/Off
POP		✓	Window Pop
PS	PE		Printer Start/End
PUSH		✓	Window Push
RB			Ring Bell
RED		✓	Red Foreground Color
RL		✓	Read Line
RP		✓	Read Page to End
RS		✓	Refresh Screen
SB	SF		Set Background/Foreground
SL	EL		Start/End Load
SWAP		✓	Window Swap

ASCII Codes

In Hexadecimal and Decimal

ASCII Control Characters

ASCII	Hex	Character
000	00H	NUL
001	01	SOH
002	02H	STX
003	03H	ETX
004	04H	EOT
005	05H	ENQ
006	06H	ACK
007	07H	BEL
008	08H	BS
009	09H	HT
010	0AH	LF
011	0BH	VT
012	0CH	FF
013	0DH	CR
014	0EH	SO
015	0FH	SI
016	10H	DLE

ASCII	Hex	Character
017	11H	DC1
018	12H	DC2
019	13H	DC3
020	14H	DC4
021	15H	NAK
022	16H	SYN
023	17H	ETB
024	18H	CAN
025	19H	EM
026	1AH	SUB
027	1BH	ESC
028	1CH	FS
029	1DH	OS
030	1EH	RS
031	1FH	US
127	7FH	DEL

ASCII Printable Characters

ASCII	Hex	Character
032	20H	SPACE
033	21H	!
034	22H	"
035	23H	#
036	24H	\$
037	25H	%
038	26H	&
039	27H	'
040	28H	(
041	29H)
042	2AH	*
043	2BH	+
044	2CH	,
045	2DH	-
046	2EH	.
047	2FH	/
048	30H	0
049	31H	1
050	32H	2

051	33H	3
052	34H	4
053	35H	5
ASCII	Hex	Character
054	36H	6
055	37H	7
056	38H	8
057	39H	9
058	3AH	:
059	3BH	;
060	3CH	<
061	3DH	=
062	3EH	>
063	3FH	?
064	40H	@
065	41H	A
066	42H	B
067	43H	C
068	44H	D
069	45H	E

070	46H	F
071	47H	G
072	48H	H
073	49H	I
074	4AH	J
075	4BH	K
ASCII	Hex	Character
076	4CH	L
077	4DH	M
078	4EH	N
079	4FH	O
080	50H	P
081	51H	Q
082	52H	R
083	53H	S
084	54H	T
085	55H	U
086	56H	V
087	57H	W
088	58H	X
089	59H	Y
090	5AH	Z
091	5BH	[
092	5CH	\
093	5DH]
094	5EH	^
095	5FH	_
096	60H	'
097	61H	a
098	62H	b
099	63H	c
100	64H	d

101	65H	e
-----	-----	---

ASCII	Hex	Character
102	66H	f
103	67H	g
104	68H	h
105	69H	i
106	6AH	j
107	6BH	k
108	6CH	l
109	6DH	m
110	6EH	n
111	6FH	o
112	70H	p
113	71H	q
114	72H	r
115	73H	s
116	74H	t
117	75H	u
118	76H	v
119	77H	w
120	78H	x
121	79H	y
122	7AH	z
123	7BH	{
124	7CH	
125	7DH	}
126	7EH	~

Error Codes

System

? 1 Directive or Function Not Available

Input/Output

00 File, Record, or Device Busy/Timeout Error

01 End of Record

02 End of File

03 Key Field Not Found

07 File Corruption Detected

File Usage

- 10 File ID Size or Key Usage
- 11 Missing or Duplicate Key
- 12 Undefined or Duplicate File ID
- 13 File/Device Access
- 14 File/Device Usage
- 15 Out of Disk Space
- 16 Disk Directory Capacity
- 17 Invalid Parameter
- 18 Illegal Program Encryption Error
- 19 Program Format or Size Incorrect

Structure

- 20 Statement Structure (Syntax)
- 21 Statement Number Error
- 22 Uninitialized Variable
- 24 Function Name Definition Already Exists
- 25 Undefined Function
- 26 Variable Usage Incorrect
- 27 Return Without GOSUB
- 28 NEXT Without FOR
- 29 Undefined Mnemonic Constant

Limit

- 30 Program Checksum Error
- 31 Internal Stack Overflow
- 32 Record Too Large for Buffer
- 33 Memory Capacity Exceeded
- 34 FOR/NEXT GOSUB/RETURN Stack Overflow
- 35 LISTer Stack Overflow
- 36 CALL/ENTER Mismatch
- 37 Format Structure/Locate Table Overflow
- 38 Illegal Command in a Public Program
- 39 Escape in a Public Program

Execution

- 40 Numeric Value Overflow

- 41 Integer Range
- 42 Nonexistent Subscript
- 43 Numeric Format Mask Overflow
- 44 Step Size Zero
- 45 Statement Usage
- 46 String Size
- 47 Invalid Substring Reference
- 48 Input Verification
- 49 Global Variable Error
- 50 Cannot Remove Primary Sort
- 51 Cannot Have More than 16 Sorts for an MSORT File
- 52 Cannot Have More than 8 Sorts for a TISAM File
- 53 Too Many Segment Definitions
- 54 Primary Key Must Be Unique
- 55 Sort Name Too Long
- 56 Field Number Greater than 255
- 57 Undefined Mode
- 58 Field Does Not Exist

Transaction Processing

- 60 Transaction Log File Not Open
- 61 Transaction in Progress
- 62 Transaction Not Started
- 63 Transaction Log Already Opened
- 64 Channel Not Opened For Transaction Modification
- 65 Transaction "IN PROGRESS" File Exists

Windows

- 70 Windows Terminal Driver Was Not Selected When This Task Started
- 71 Windows Error: Too Many Active Windows or Panels
- 72 Windows Error: Attempt To Delete or Save the Main Window
- 73 Windows Error: Illegal Value in Attribute Map or String
- 74 Windows Error: The Windows System is Disabled
- 75 Windows Error: Illegal Length or Value for Window Name
- 76 Windows Error: Illegal String Parameter Length
- 77 Windows Error: Illegal Numeric Parameter Value

- 78 Windows Error: Wrong Format or Length for Command Option
- 79 Windows Error: Illegal Window Command Option Keyword
- 80 Windows Error: Attempt to Use the Same Optional Parameter Twice
- 81 Windows Error: Non-keyword = Option Used as Keyword = Option
- 82 Windows Error: Illegal Window Command Option Keyword Value
- 83 Windows Error: Border Character Must Be a Printable Character
- 84 Windows Error: Illegal Format or Value for Border Attribute
- 85 Windows Error: Illegal Window Command Option Keyword Value
- 86 Windows Error: Illegal Window Command Option for This Command
- 87 Windows Error: New Window or I/O Region Will Not Fit
- 88 Windows Error: Undefined or Duplicate Window Name
- 89 Wrong Format for Window Contents Map(s)
- 90 Windows Error: Map Length Wrong for This Window or I/O Region
- 91 Windows Error: Illegal Map Type
- 92 Windows Error: Unprintable Character in Text Map or String
- 93 Windows Error: I/O Region Column and/or Row Count Is Zero
- 94 Windows Error: Illegal Coordinate and/or Length for Get/Put Command
- 95 Windows Error: Cannot Change a Function Key that Does Not Exist

Other

- 103 Unexpected Operating System Error
- 127 Not an Error (the Escape Key sets ERR=127)

Dictionary-IV Interface

- 160 Program Contains Invalid Format/Data Name References
- 161 Undefined Format Name
- 162 Format Name Has Not Been INCLUDED
- 163 Undefined Data Name
- 164 Data Name Does Not Allow Multiple Occurrences
- 165 Invalid Occurrence

- 166 String/Numeric Mismatch of Data Name
- 167 Invalid Value for Data Name
- 168 Undefined Data Dictionary
- 169 Format Corruption Detected
- 170 Format Cannot Be Deleted (OPEN LINK)
- 171 Undefined Link Name
- 172 Cannot Process a Format/Data Name in the IOLIST of an
OPEN LINK

Notational Symbols

Syntax appears in a bold non-proportional font to better display spaces and punctuation. Elements of the syntax are always explained directly below. The following conventions are used in command formats throughout the Thoroughbred product line.

(BOLD FACE/UPPERCASE)

CONNECT VIEW

Commands or keywords you must type exactly as shown.

Example: **CONNECT VIEW UTVCUST**

You must type CONNECT VIEW including the space.

UPPERCASE ITALIC FACE

BUSY PROCESS IS procedure

Optional words used for script readability and have no effect

on command syntax.

Example: ***BUSY PROCESS IS procedure***

Italic Face

view-name-1

Information you must supply.

Example: **CONNECT VIEW UTVCUST**

You supply the view name UTVCUST.

In most cases, lowercase italics denotes values that accept uppercase or lowercase characters.

Underscore

Option-1 | Option-2 | Option-3

Either a default in a command description or display default in a screen image.

Example: **Add | Change | Delete**

You may select the Add, Change, or Delete option. The Change option is the default value.

Braces { }

{view-name-1} [[view-name-2]...]

You must select one of the values enclosed by the braces.

Example: **{UTVCUST1} [[UTVCUST2]...]**

You must supply the view name UTVCUST1.

Brackets []

{view-name-1} [[view-name-2]...]

You can select one of the options enclosed by the brackets;

none of the enclosed values is required.

Example: **{UTVCUST1} [[UTVCUST2]...]**

You must supply the view name UTVCUST1.

You may also supply the optional view name UTVCUST2.

Ellipsis ...

{view-name-1} [[view-name-2]...]

You can repeat the word or clause that immediately precedes the ellipsis.

Example: **{UTVCUST1} [[UTVCUST2]...]**

You must supply the view name UTV CUST1.

You may also supply the optional view names UTV CUST2, UTV CUST3, UTV CUSTX. X

represents any existing view name in the library.

Vertical Bar |

Option-1 | Option-2 | Option-3

Piping separates options. One vertical bar separates options, two vertical bars separate three options. You

select

only one of the options.

Example: **Add | Change | Delete**

You may select the Add, Change, or Delete option.

Punctuation and Symbols

With the exception of the above symbols, all punctuation or relational symbols shown within a command format, such

as

commas, parenthesis, semicolons, equal signs, etc. are part of the syntax and must be included.

Example: **DO procedure-0 [,procedure-1**

[,procedure-2] ...]

You must use the comma between procedures.

Software Conventions

Function Keys

- F4** exit.
- F6** on-line help.

Function Keys from the Definition Name Field

- F1** switch maintenance modes.
- F2** display lookup of definition names.
- F4** display the class selector.

Cursor Movement During Field Edit

Left/Right Arrow

Moves the cursor one character left or right in the field. The cursor does not exit the field until you exit the edit mode.

Tab/Back Tab

Moves the cursor half the width of the input window to the right or left in the field. The cursor does not exit the field until you exit the edit mode. In this example 23 characters is half the width of the window.

Home

If the cursor is on the first character in the field, the system moves it to the last position.
If the cursor is on any other character in the field, the system moves it to the first position.

Character Insert/Delete

Press **Character Insert** in a field to insert characters in the field without typing over any characters.

Press **Character Delete** in a field to delete characters from the field.

Line Insert/Delete

Press **Line Insert** to insert a row in a file.

Press **Line Delete** to delete a row in a file.

Clear Line

Deletes all the characters from the field.

Types of Menus

IDOL-IV

Displays options. Type the selection number and press **Enter**.
For example, UTMENU10.

Pop-up

Allows you to scroll with a lightbar to highlight an item and press **Enter** to select it. For example, press **F1** from any IDOL-IV menu to display a pop-up menu.

Toggle

Allows you to select one or more items on the menu. Press the **Space Bar** or **Enter** to toggle. You must select **Execute** to exit and save the changes. For example, Press **15** in a CONNECT VIEW.

Using IDOL-IV Menus

Action	Description
F1	Displays the class pop-up menu for definition maintenance.
F2	Displays the Screen/View database maintenance pop-up menu.
F3	Displays the Thoroughbred product pop-up menu.
F6	Displays on-line help for the menu selection.
?F6	Displays on-help for menu selections.
#F6	Displays on-line help for the specified menu selection.
F7	Displays the language code selections.
F10	Displays the menu names.
Ctl-P	Activates Desk Services.
.query	Runs the specified query.
.report	Runs the specified report.
;menu	Displays the specified menu.
/program	Runs the specified program.
:program	Runs the specified program.
DBM	Displays the Screen/View database maintenance pop-up menu.
DM	Displays a lookup of menu names.
ID	Displays the IDOL-IV Development Menu.
LOF	Logs off and returns to the Enter Operator Code screen.
MAIN	Returns to the logon menu and clears the menu trail.
MRM	Allows you to perform multi-record maintenance on the selected View, Link, or Format.
MS	Accesses the internal mail system.

SRM	Allows you to perform single-record maintenance on the selected Screen,, Link,, or Format.
TQ	Runs <i>Query-IV</i> .
TS	Runs <i>Source-IV</i> .

Printing

If you select a print option or press **F16**, IDOL-IV prompts:

Select Printer (**xx**, **x₁x₁**, **x₂x₂**):

xx, x₁x₁, x₂x₂

refers to the names of the printers you added during Printer Assignment. For more information see Printer Table Maintenance and Printer Assignment in the *Dictionary-IV System Administrator Guide*.

Type the printer name and press **Enter**. You can press **F1** for a lookup of printer names.

Common Syntax Elements

Some commands in this manual use common syntax elements, which are explained in this section.

Expressions

An expression is two or more elements, consisting of constants, element names, variables, or functions that interact with operators to form a new value.

An expression can be used in a screen data field (formula), and conditions.

Operators in Expressions

+	addition or string concatenation
-	subtraction
*	multiplication
/	division
^ or **	exponentiation
()	grouping

Numeric expressions are evaluated according to the normal convention (grouping, exponentiation, multiplication/division, addition/subtraction).

Logical Conditions

A logical condition is two or more values, consisting of constants, data element names, variables, functions, or expressions that interact with relational or logical operators to form either a true or false result.

Operators in Conditions

=	equal to
>	greater than
<	less than
>= or =>	greater than or equal to
<= or =<	less than or equal to
<> or ><	not equal to
()	grouping
AND	logical AND (both true)

LIKE search for a match using partial values
OR logical OR (either true)

Special Operators in Conditions

NOT value

Negative condition. The NOT operator is used to negate a condition. It can be used to negate the result of the IN, BETWEEN, or LIKE operators.

IN (constant [,constant]...)

Matching value in list. Items in the list specified by the IN operator must each be separated by a comma and space. Alphanumeric values must be padded to the correct length for a proper match.

LIKE "partial-value"

Partial equality. The LIKE operator can specify a string value containing wildcards, which can match more than 1 character. LIKE automatically pads its values to the correct length. (You must manually pad values compared by other operators.)

LIKE Wildcards

- * Matches any characters (0 or more)
- ? Matches a single character
- [A-Z] Matches a range for a single character
- [AGCF] Matches a single character in list
- [wildcatted] Matches the wildcard character

The brackets [] in wildcards are required; they do not indicate optional values. The * and ? wildcards do case-insensitive comparisons.

BETWEEN low-value AND hi-value

Greater than or equal to and less than or equal to. This operator can work with numeric or alphanumeric values, but the low -value and hi-value must be of the same type. Alphanumeric values must be padded to the correct length if you want to obtain a specific match for starting and ending values.

RANGE FROM "low-value" TO "hi-value"

Primary key field greater than or equal to and less than or equal to. This operator works only with alphanumeric values. It uses the primary key field from the first link defined in the FROM clause to select a range of records. The key field does not have to be used as a column in the query. Alphanumeric values must be padded to the correct length if you want to obtain a specific match for starting and ending records.

SOUNDEX value

Sounds like. The SOUNDEX operator will compare values using a phonetic operation to find data that has a similar pronunciation.

Boolean operators

All Boolean operators can be used in arithmetic functions.

Operators In Conditions

If the relation is true, the result is 1; otherwise, the result is 0.

Logical Operators

The result of AND is 0 if either or both of its operands are 0; otherwise, the result is 1. The result of OR is 0 if both of its operands are 0; otherwise the result is 1. If more than one logical operator appears in a statement, and the result is determined after evaluating the first operand, the statement will "short circuit", i.e., skip the evaluation of the second operand.

Examples:

```
00010 LET A=0, B= 0, C=5, D= -4, E=7, F=2
00020 LET VALUE= A+C OR B ;VALUE IS 1
00030 LET VALUE = A AND B ;VALUE IS 0
00040 PRINT (C AND E) + 1 ;PRINTS 2
00050 LET VALUE= A=B ;VALUE IS 1
00060 PRINT (D>F OR E<C) +10 ;PRINTS 10
00070 IF A THEN GOTO 10
00080 LET VALUE = A$ LIKE A?PSD@
```

Any numeric syntax that is accepted between IF-THEN can be used as a valid numeric expression anywhere a number is expected or allowed.

String

A string is a connected sequence of characters treated as one piece of data.

Substring

Substrings can be used in screen data fields, sort definitions, and view commands. They allow you to specify a portion of the original value in a data element name.

- ▶ Substrings can only be used with alphanumeric data elements.
- ▶ Substrings cannot be used with a multiple occurrence element.

The syntax is as follows:

```
string-elmt[(start,length)]
```

string-elmt is an alphanumeric data element name.

start is the starting position for the substring.

length is the number of characters of the substring.

Masks (Output Format)

Syntax

The syntax is as follows:

```
num-elmt:"mask"
```

num-elmt is a numeric data element name, formula, or multiple occurrence.

mask is one or more mask characters.

Mask Characters (Numeric Output)

An insignificant zero is a leading or trailing zero. A floating character is output at the rightmost leading zero.

0	Outputs a digit from 0 to 9. When the digit is an insignificant zero, outputs a 0.
#	Outputs a digit from 0 to 9. When the digit is an insignificant zero, outputs a space.
.	Outputs a decimal point between digits. If the number is zero and 0 is not used in the mask, outputs a space.
,	Outputs a comma between digits if the digit to the left is a significant digit; can only be used to the left of the decimal point.
\$	Outputs a dollar sign.
*	Outputs an asterisk when the digit is a leading zero.
-	Outputs a minus sign if the number is negative, or a space if the number is positive. This mask character can be placed at the left or right of the mask. If placed to the left, it is a floating character.
+	Outputs a plus sign if the number is positive, a minus sign if the number is negative or a space if the number is zero. This mask character can be placed at the left or right of the mask. If placed to the left, it is a floating character.
CR	Outputs a CR (to indicate a credit) if the number is negative or 2 spaces if the number is positive. This mask character must be placed at the right of the mask.
DR	Outputs DR (to indicate a debit) if the number is positive, a CR if the number is negative or 2 spaces if the number is zero. This mask character must be placed at the right of the mask.
(Outputs a left parenthesis if the number is negative or a space if the number is zero or positive. This mask character must be placed at the left of the mask.
)	Outputs a right parenthesis if the number is negative or a space if the number is zero or positive. This mask character must be placed at the right of the mask.
B (Blank)	Outputs a space.
????	Any characters other than the mask characters are output in the specified position.

Date Masking Characters

Date masking characters can be used as valid mask characters with SQL numeric dates.

YY	Two-digit year, showing only the least significant 2 digit of the year.
YYY	Three-digit year, showing only the least significant 3 digits of the year.
YYYY	Full four-digit year; leading minus sign will appear for BC dates.
MM	Two-digit month (e.g. January = 01 December =12).
MON	Uppercase three-character abbreviation of the month (e.g. JAN).
MONTH	Full uppercase name of the month (e.g. JANUARY).

Mon	Upper/lower-case three-character abbreviation of the month (e.g. Jan).
Month	Upper/lower-case full name of the month (e.g. January).
DD	Two-digit day of the month.
DDD	Three-digit day of the year (Julian) from 001 through 366.
DY	Uppercase 3 character abbreviation of the day of the week (e.g. MON).
DAY	Uppercase full day of the week (e.g. MONDAY).
Dy	Upper/lower-case 3 character abbreviation of the day of the week (e.g. Mo).
Day	Upper/lower-case full day of the week (e.g. Monday).
HH	Hour of the day in 24-hour format.
MI	Minutes.
SS	Seconds.

Services

The following sections describe the services provided by Thoroughbred Software International, Inc.:

Printed Documentation

Thoroughbred provides printed documentation for its products. Refer to the Product Reference Manual or your Thoroughbred Sales Representative for a list of available documentation.

On-line Documentation

Thoroughbred provides on-line documentation for IDOL-IV products. You must have installed the Developer Reference module when you installed IDOL-IV. For more information see the IDOL-IV Installation Guide.

You may access help by pressing **Ctrl-P** and selecting Help Topics or by pressing **F6** at any menu option or any field.

The Help Topics option allows you to search the help by a keyword or phrase. For example, if you need help for the Sort File field in Link Definition type **Sort File** and press **Enter**. The system displays all matching sort file topics.

If you are operating in a Windows 95 or Windows NT environment, Thoroughbred Software International, Inc. provides Windows help. You may access this help by selecting the Help file you want from **My Computer** or **Windows Explorer**.

Support

Thoroughbred offers customer support by telephone, fax, CompuServe, and e-mail. Our e-mail address for customer support is **support@tbred.com**.

On-line Communication

Thoroughbred is active on the CompuServe UNIX Vendor Forum. To access the forum select the **Go** option from the Services menu, type **tbred**, and press **Enter**.

Thoroughbred provides information and services on the Internet. Please visit Thoroughbred on the World Wide Web at **<http://www.tbred.com>**.

Thoroughbred publishes information and responds to comments and questions through these services. It provides a 24 hour reference and communications vehicle for all Thoroughbred developers and end-users worldwide.

Training Classes

Thoroughbred offers training classes on IDOL-IV and other products. These classes are held at the corporate headquarters as well as regional locations. For more information and a schedule of upcoming classes, please contact the Thoroughbred Training Department at 1-800-524-0430 or 908-560-1377.