Thoroughbred® Basic™
Customization and Tuning Guide

Volume II: System Files
Version 8.6.0

285 Davidson Ave., Suite 302 • Somerset, NJ 08873-4153
Telephone: 732-560-1377 • Outside NJ 800-524-0430
Fax: 732-560-1594

Internet address: **http://www.tbred.com**

Published by:
Thoroughbred Software International, Inc.
285 Davidson Ave., Suite 302
Somerset, New Jersey 08873-4153

Document Number: BC8.6.0M101

Preface

After you install or upgrade Thoroughbred Basic you can modify and optimize it to fit smoothly within the environment defined by your hardware and operating system. As you develop, maintain, and enhance applications the Thoroughbred Basic Customization and Tuning Guide will provide information that helps you tailor Thoroughbred Basic to meet current and future site requirements.

If you have just finished an installation you may want to define site terminals and printers to Thoroughbred Basic before you release the product to your user community. After an installation or upgrade you may want to skim the subjects covered in this manual as a way of helping you devise a development environment that will make the most of site hardware and firmware.

This manual assumes familiarity with UNIX concepts and commands. Experience as a system administrator or Thoroughbred Basic programmer will help facilitate customization and tuning tasks.

The Thoroughbred Basic Customization and Tuning Guide is contained in two volumes. Volume I describes how to establish terminals, printers, directories, and ghost tasks to Thoroughbred Basic. Volume II describes how to use system files to make full use of site hardware and firmware.

This manual is a companion to the Thoroughbred Basic Installation and Upgrade Guide and part of a Thoroughbred Software International documentation library that includes the Thoroughbred Basic Developer Guide, the Thoroughbred Basic Language Reference, the Thoroughbred Basic Utilities Manual, the Thoroughbred Basic Technical Appendices, and the Thoroughbred Basic Quick Reference Guide.

For more information on how to customize and tune Thoroughbred Basic under Microsoft Windows, please refer to the Thoroughbred Environment for Windows Installation and Upgrade Guide.

Notational Symbols

BOLD FACE/UPPERCASE	Commands or keywords you must code exactly as shown. For example, **CONNECT** VIEWNAME.
Italic Face	Information you must supply. For example, **CONNECT** *viewname*. In most cases, *lowercase italics* denotes values that accept lowercase or uppercase characters.
UPPERCASE ITALICS	Denotes values you must capitalize. For example, **CONNECT** *VIEWNAME*.
Underscores	Displays a default in a command description or a default in a screen image.
Brackets [ ]	You can select one of the options enclosed by the brackets; none of the enclosed values is required. For example, **CONNECT** [VIEWNAME|*viewname*].
Vertical Bar |	Piping separates options. One vertical bar separates two options, two vertical bars separate three options. You can select only one of the options
Braces { }	You must select one of the options enclosed by the braces. For example, **CONNECT** {VIEWNAME|*viewname*}.
Ellipsis **. . .**	You can repeat the word or clause that immediately precedes the ellipsis. For example, **CONNECT** {*viewname1*}[[**,** *viewname2*] **. . .**].
lowercase	displays information you must supply, for example, SEND filename.txt.
Brackets **[ ]**	are part of the syntax and must be included. For example, SEND **[**filename.txt**]** means that you must type the brackets to execute the command.
punctuation	such as **,** (comma), **;** (semicolon), **:** (colon), and **( )** (parentheses), are part of the syntax and must be included.

1. System Files

Thoroughbred Basic can run under a variety of hardware and firmware environments. To insure portability across hardware and firmware platforms, the Thoroughbred Basic system insulates applications developed in Thoroughbred Basic from environmental concerns. If an application does not explicitly depend on a given hardware and software configuration, you can easily port the application to a new platform.

However, the Thoroughbred Basic system is not free from environmental concerns. It must interact with a given set of hardware and firmware. This chapter describes how Thoroughbred Basic works with the resident operating system to obtain the services it needs and how to use system files to make full use of hardware and firmware:

- For a brief description of operating system services that Thoroughbred Basic uses, and a discussion of how Thoroughbred Basic runs under an operating system that uses windows, please refer to Chapter 2, **Environmental Considerations**.

- The TCONFIGW, TCONFIG8, and TCONFIG files contain terminal specifications that describe how a terminal will interact with Thoroughbred Basic and the resident operating system. For more information on these files, please refer to Chapter 3.

- The TERMINAL file contains a list of the port name, operating system window, or network node name for each Thoroughbred Basic task. For a description of the TERMINAL file, please refer to Chapter 4.

- The TERM.MAP file enables you to associate terminal definitions with task IDs. For a description of the TERM.MAP file, please refer to Chapter 5.

- The TSI.INI file is where you enable or disable features available in the Thoroughbred Environment for Windows. For more information on this file, please refer to Chapter 6.

- The SERVER.MAP file enables you to establish the interface between Thoroughbred products and Oracle databases, enables you to establish the interface between Thoroughbred products and ODBC-compliant databases, or enables new or existing applications to share data across a network. For a description of the SERVER.MAP file, please refer to Chapter 7.

- The IPLINPUT file is an initial program load (IPL) file that contains environmental information. IPLINPUT is loaded into memory whenever a user starts Thoroughbred Basic unless the user specifies a different IPL file. For more information on the IPLINPUT file, and for more information on how to create customized IPL files, please refer to Chapter 8.

2. Environmental Considerations

Thoroughbred Basic can run under an operating system or an operating environment generated by an operating system. This section describes the operating system services used by the Thoroughbred Basic system and the applications created under this system, and it discusses how to fit the system within an operating system environment that makes use of windows.

2.1 Operating system services

The Thoroughbred Basic system, and the applications developed under this system, can request services from the resident operating system. The four major types of operating system services are:

Memory Management Services

Thoroughbred Basic requests Random Access Memory (RAM) from the operating system to store programs and data.

In most cases, these requests are implicit. In cooperation with Thoroughbred Basic, the operating system manages the allocation and de-allocation of RAM.

Mass Storage Services

Thoroughbred Basic uses operating system facilities to save, retrieve, and manage program files and data files.

In most cases, requests for storage need minimal information, such as the name under which a new file will be saved and the disk where the file will be placed. Operating system facilities allocate space, update file tables, and so on.

Communication Port Services

Thoroughbred Basic uses these services to communicate with peripheral devices such as printers, modems, other computer systems, point-of-sale devices, and plotters.

In most cases, requests for communication port services need minimal information, such as the name of a port or a terminal. Operating system facilities manage most data transmission issues.

Task Management Services

Thoroughbred Basic relies on the operating system to control data flow to and from individual tasks. In most cases, task management means managing terminal communications. However, task management can extend to a task that does not have its own terminal, such as a ghost task or a task driven from a remote computer.

Part of this service is the recognition of independent tasks and their rights to restrict access to individual data records and data files. These locking facilities are necessary in a multi-user environment.

In most cases, Thoroughbred Basic issues implicit requests for task management services.

2.2 Windowing systems

Many operating systems provide a user interface manager, which can be implemented in a variety of ways. A user interface manager can be a hotkey service facility or a graphical user interface (GUI).

Thoroughbred Basic does not prevent you from using these facilities. In most cases, it will run under user interfaces that are application-independent, so if a Thoroughbred Basic application does not need to know that the interface exists, the application will run.

Some user interface managers implement a graphical user interface as a windowed operating environment. A terminal that can display multiple windows can run multiple Thoroughbred Basic tasks by starting Thoroughbred Basic in each window. If you plan to run Thoroughbred Basic under a windowed operating environment, you may need to modify the TERMINAL file.

Each Thoroughbred Basic terminal requires a unique Thoroughbred Basic task ID, which is also called a terminal ID. In many windowed environments, each open window can be regarded as a terminal. Each window that contains an instance of Thoroughbred Basic requires a unique task ID.

In this case, you must make sure that the system name for each window that runs Thoroughbred Basic is specified in the TERMINAL file. For example, imagine that want two windows on one terminal to concurrently run Thoroughbred Basic. If the operating system window manager calls the first window **tty01a** and the second **tty01b**, you must specify both names in the TERMINAL file.

For more information on the TERMINAL file, please refer to Chapter 4.

3 The TCONFIG Files

These Thoroughbred Basic files contain specifications and definitions that enable Thoroughbred Basic to manage terminal hardware. You can create or modify a set of terminal specifications, which is called a terminal table.

Terminal tables are necessary because terminal manufacturers use different sets of standards to build their terminals. Most terminals use the same character codes to specify printable characters. For example, a hexadecimal value of 20 usually specifies a space character and a hexadecimal value of 41 usually specifies the A character. However, the character codes used to specify unprintable characters, control sequences, edit codes, and function key codes vary widely throughout the industry.

You can use the terminal tables in the TCONFIG files to specify character codes for unprintable characters and provide the interface that will allow Thoroughbred Basic to make full use of terminal capabilities. Information contained in the TCONFIG files includes:

- The terminal tables Thoroughbred Basic requires for terminal input and output. The tables are listed by manufacturer name and model number.

- A list that specifies the terminal table Thoroughbred Basic will associate with each task ID. The task ID, which is also called the terminal ID, is assigned by the position of the system port name in the TERMINAL file. For more information on the TERMINAL file, please refer to the following section.

The TCONFIG files are called TCONFIGW, TCONFIG8, and TCONFIG:

- The **TCONFIGW** file enables Thoroughbred Basic to make use of Dictionary-IV and Thoroughbred Basic Windows.

  Thoroughbred Dictionary-IV is the data dictionary component of the Thoroughbred IDOL-IV application development system. It can provide many resources to Thoroughbred Basic. For more information on these resources please refer to the chapter on the Dictionary-**IV Interface** in the Thoroughbred Basic Developer Guide.

  Thoroughbred Basic Windows is a windowing system that enables you to create pop-up help systems, menu systems, and other program features commonly associated with windowing systems. For more information on Thoroughbred Basic Windows, please refer to the Thoroughbred Basic Developer Guide.

  If you have terminals with graphics capabilities, if you plan to use Dictionary-IV, or if you plan to develop applications that make use of the Thoroughbred Basic Windows graphical user interface, you must use this file.

  The TCONFIGW file is available starting with Thoroughbred Basic 8.1.

- The **TCONFIG8** file does not enable you to use Dictionary-IV or Thoroughbred Basic Windows.

  If you have terminals that do not support graphics functions, if you do not plan to use Dictionary-IV, or if you plan to develop applications that are text-based and linear, you may need to use this file.

  The TCONFIG8 file is available starting with Thoroughbred Basic 8.0.

- The **TCONFIG** file does not enable you to use Dictionary-IV or Thoroughbred Basic Windows. The TCONFIG file is available in releases prior to Thoroughbred Basic 8.0.

Thoroughbred Basic requires you to use at least one of these files. The TCONFIG files are placed in the **/UTILS** directory under the Thoroughbred Basic home directory. You may not have all three of these files.

To modify the TCONFIGW or TCONFIG8 file, you can use the information and procedures described in Chapter 2 in Volume I of the Thoroughbred Basic Customization and Tuning Guide. The **install** utility described in that chapter uses the ***NPSD** utility to modify these files. For more information on the ***NPSD** utility, please refer to the Thoroughbred Basic Utilities Manual.

4. The TERMINAL File

The TERMINAL file is an operating system file. It contains a list of the port names, operating system window names, or network node names for each task that can execute Thoroughbred Basic. The names specified in this list are names used and assigned by the operating system. For example, in UNIX, **console** may be the port name for the system console and **tty05a** may specify an operating system window. On a network running under MS-DOS, **000000000004** may specify a network node.

Thoroughbred Basic then assigns a unique name, called a task ID or a terminal ID, to each name in the TERMINAL file. Each task ID is unique within the Thoroughbred Basic system and the operating system.

Thoroughbred Basic assigns the **T0** task ID to the first name in the TERMINAL list, **T1** to the second name, and so on. The naming sequence runs from **T0** through **T9**, then **TA** through **TZ**, then **Ta** through **Tz**, then **U0** through **Uz**, and so on. Valid values for the first letter are **T**, **U**, **V**, **W**, **X**, **Y**, **Z**, **A**, **B**, **E**, **H**, **I**, **J**, **K**, and **M**. Thoroughbred Basic can assign up 930 task IDs.

The letters not listed above are used to form name for special classes of system tasks or devices. For example:

D is used to name logical disk directories.

G is used to name ghost tasks.

L is used to name printers.

P is used to name printers.

Because Thoroughbred Basic uses 2-character names to specify tasks, devices, and operating system services, programmers should not specify 2-character names for files created under Thoroughbred Basic. In some cases, Thoroughbred Basic can confuse 2-character file names with 2-character task IDs. For example, on a system with a floppy disk drive specified as **F0** and a file named **F0**, a request for information issued from Thoroughbred Basic may not retrieve the information the programmer requires.

The Thoroughbred Basic system uses the information in the TERMINAL file when a user starts Thoroughbred Basic:

./b [*taskID*] [*IPLfile*]

./ tells the operating system to search the current directory, in this case the Thoroughbred Basic home directory.

b is the command to start Thoroughbred Basic under UNIX.

taskID is the name of the task ID, which is also called the terminal ID. In a multi-user environment, all of the users at your site must agree to start Thoroughbred Basic in just one of the following ways:

- Allow Thoroughbred Basic to assign task IDs, in which case no user specifies a value for *taskID*. A task ID is automatically assigned to a terminal depending on the position of the terminal device name in the TERMINAL file.

- Use a unique task ID, in which case every user specifies a value for *taskID*. Because task IDs are associated with terminals, a user who works at several terminals may have to remember several task IDs.

  Site procedures must specify just one of these ways to start Thoroughbred Basic. If both methods are used, more than one user may use the same task ID, which can cause problems such as file corruption or display problems. For more information on site login procedures, please refer to Section 2.4.

IPLfile is the name of the initial program load (IPL) file, which contains specifications that are automatically implemented and commands that are automatically executed when a user starts Thoroughbred Basic. The default is IPLINPUT, a file in the Thoroughbred Basic home directory.

For more information on the IPLINPUT file, please refer to the section on **The IPLINPUT file** in this manual.

If you plan to modify the TERMINAL file, you can use a text editor such as vi, ed, or edlin. For more information on valid system names, please refer to your operating system documentation.

5. The TERM.MAP File

This Thoroughbred Basic file enables Thoroughbred Basic to associate terminal definitions with task-IDs. As an alternative to the standard terminal configuration method for associating a terminal table with a task ID, you can select a terminal table before you execute Thoroughbred Basic.

You can create the TERM.MAP file in the Thoroughbred Basic home directory and assign a permission set of rw-rw-rw to the new file. Then, you can use the following format to create records to specify terminal types:

V1[|*V2*|**. . .** *Vn*]**:***table*

V1	is a terminal specification recognized by the operating system. For example, your operating system may recognize W60E-25 as a specification for the Wyse 60 terminal.
|	is the pipe character. This character separates each terminal specification.
V2|**. . .** *Vn*	are alternate specifications for the terminal specification contained by *V1*.
:	is a colon.
table	is the Thoroughbred Basic terminal table name specified in the TCONFIGW, TCONFIG8, or TCONFIG file.

Example

W60E-25|**wy60-15**|**w601**|**wyse60-25**:**W60E**

The example above specifies a set of recognized names for the Wyse 60 terminal and associates all of those names with the Thoroughbred Basic W60E terminal table.

You can find valid terminal designations in the file where your operating system stores information on terminal characteristics. Many UNIX systems store this information in the **/etc/termcap** or **/etc/terminfo** file.

You can specify the appropriate terminal table in the operating system by setting the **TERM** variable before you execute Thoroughbred Basic. For example, from your operating system prompt you can type:

TERM=vt220; export TERM

and press the **Enter** key. When you execute Thoroughbred Basic, it gets the contents of the **TERM** variable and checks TERM.MAP, which provides the name of a terminal table that the TCONFIG file will use.

TERM.MAP entries take precedence over the ***NPSD** or **8NPSD** defaults. However, you can still use the ***NPSD** or **8NPSD** utility to activate and load a terminal table specified in a TCONFIG file.

For more information on the ***NPSD** utility, please refer to the Thoroughbred Basic Utilities Manual. For more information on the **8NPSD** utility, please refer to the Dictionary-IV Reference Manual. For more information on the standard method used to associate terminal tables with task IDs, please refer to Section 2.3 in Volume I of the Thoroughbred Basic Customization and Tuning Guide.

6. The TSI.INI File

The TSI.INI file contains three sections: [TIS], [BASIC] and [MOUSE]. The [TSI] section is used by install programs and should not need to be modified. The [BASIC] and [MOUSE] sections may be edited to set your own preferences. The options available are summarized below and discussed in detail in Section 6.1 and 6.2.

```
[BASIC]
window_name=<Insert window name here>
window_title=<Insert window title here>
window_icon=<Insert icon file name here>
window_hscroll=yes/no
window_vscroll=yes/no

[MOUSE]
mouse=on/off
click=on/off
dblclick=on/off
rightclick=on/off
controlkey=<>,F-,F=
yesno=(Y/N)?
smartcopy=on/off
```

6.1 The [BASIC] Section

window_name=

Allows you to change the name displayed in the title bar of the Basic window.

window_title=

Allows you to change the title displayed in the title bar of the Basic window. Please note that this option is not available in the Windows Pocket PC Basic.

window_icon=

Allows you to change the icon displayed in the title bar of the Basic window. Please note that this option is not available in the Windows Pocket PC Basic.

window_hscroll=

Allows you to turn on or off horizontal scrolling of a window. Please note that this option is only available in the Windows Pocket PC Basic.

window_vscroll=

Allows you to turn on or off vertical scrolling of a window. Please note that this option is only available in the Windows Pocket PC Basic.

6.2 The [MOUSE] Section

mouse=

Enable (on) or disable (off) the detection of the mouse events.

Note: If this option is disabled, the other mouse items will be ignored.

click=

Enable (on) or disable (off) the use of the left mouse button.

dblclick=

Enable (on) or disable (off) the use of a double-click of the left mouse button.

rightclick=

Enable (on) or disable (off) the use of the right mouse button.

smartcopy=

Enable (on) or disable (off) the smart copy feature, which determines how to highlight multiple lines of text. When smart copy is disabled, the selection of multiple lines of text will wrap around the end of each line. When smart copy is enabled, the selection of multiple lines of text will not wrap around the end of each line, allowing the selection of a column.

To copy text, drag the mouse to highlight a block of text. From the Basic Menu Bar select **Edit** and then **Copy**. The text is copied to the clipboard and is available to all window applications supporting paste from the clipboard.

To paste text in the Basic window, position the cursor where the text is to be pasted and then from the Basic Menu Bar select **Edit** and then **Paste**.

controlc=

Enable (on) or disable (off) the use of <CTL-C> to copy text.

controlv=

Enable (on) or disable (off) the use of <CTL-V> to paste text.

controlkey=

Control keys are identified by a starting and ending character, for example type "<>" in the Control Key text box. Function keys can also be identified by a starting character of "F" and an ending character, for example "F-". When the user clicks in an area encompassed by the start and end characters, the text will be interpreted. If the text contains a known control key code then that control key will be posted to the Basic INPUT statement.

Multiple Control Key values can be defined. Use a comma to separate entries. For example, type "<>", "()","{}" "<>,(),{},F-,F=" in the Control Key text box.

For example if you type "<>" in the Control Key(s) text box, clicking on the "F" or the "4" in <F4> returns a CTL value of 4.

For example if you type "F-" in the Control Key(s) text box, clicking on the "F" or the "4" in F4-End returns a CTL value of 4.

yesno=

A starting, ending and separator character identifies the Yes and No characters, for example (Y/N).

(is the start character

) is the end character

/ is the separator character

When the user clicks in an area encompassed by the starting and separator character or the separator and ending character, the text will be interpreted. If the text matches the yes/no pattern, the clicked character will be posted to the Basic INPUT statement. Only a single character is sent to INPUT, this is equivalent to an INPUT SIZ=1.

For example if you type "(Y/N)" in the Yes/No Pattern text box, clicking on the “Y” in (Y/N)? returns a Y. Clicking on the “N” (Y/N)? returns an N.

You can set your system to scroll and to enable mouse clicks by editing the TSI.INI file located in your Windows folder.

To enable/disable scrolling edit the [BASIC] section of the TSI.INI file.

window_hscroll=Y
Allows you to turn on or off horizontal scrolling of a window.

window_vscroll=Y
Allows you to turn on or off vertical scrolling of a window.

For example:

```
[BASIC]
window_hscroll=Y
window_vscroll=Y
```

To enable mouse clicks to select Y/N and Control Keys edit the [MOUSE] section of the TSI.INI file.

Yes/No Pattern
A starting, ending and separator character identifies the Yes and No characters, for example (Y/N).

(is the start character
) is the end character
/ is the separator character

When the user clicks in an area encompassed by the starting and separator character or the separator and ending character, the text will be interpreted. If the text matches the yes/no pattern, the clicked character will be posted to the Basic INPUT statement. Only a single character is sent to INPUT, this is equivalent to an INPUT SIZ=1.

For example if you type "(Y/N)" in the Yes/No Pattern text box, clicking on the "Y" in (Y/N)? returns a Y. Clicking on the "N" (Y/N)? returns an N.

Control Keys
Control keys are identified by a starting and ending character, for example type **"<>"** in the Control Key text box. When the user clicks in an area encompassed by the start and end characters, the text will be interpreted. If the text contains a known control key code then that control key will be posted to the Basic INPUT statement.

Multiple Control Key values can be defined. Use a comma to separate entries. For example, type **"<>", "()","{}"** in the Control Key text box.

For example if you type "<>" in the Control Key(s) text box, clicking on the "F" or the "4" in <F4> returns a CTL value of 4.

For example:

```
[MOUSE]
yesno=(*/*)?
controlkey=<>
```

7. The SERVER.MAP File

This system file helps establish an interface between Thoroughbred products and other databases, or enables new or existing applications to share data across a network:

- For an overview of how the SERVER.MAP file works with the TS Oracle DataServer product, please refer to Section 7.1.
- For an overview of how the SERVER.MAP file works with the TS Network DataServer product, please refer to Section 7.2.
- For an overview of how the SERVER.MAP file helps establish an interface and connection between Thoroughbred products and ODBC-compliant databases, please refer to Section 7.3.

7.1 TS Oracle DataServer

The SERVER.MAP file helps establish the interface between Thoroughbred products and Oracle databases. It establishes a relationship between a two-character server ID used by Thoroughbred products and Oracle servers.

If the file does not exist, you can create it as a simple ASCII file. If SERVER.MAP is not located in the current directory, the system searches the **/usr/lib/basic** directory, which is the default.

Entries in the SERVER.MAP file are formatted in the following way:

*server-ID***:**{*host-name*|*TCP/IP-address*}[**:***TCP/IP-port*][:*data-source*]

server-ID is a two-character ID that refers to a server.

: separates the *server-ID* from the *data-source*.

host-name is the host name of the server system.

data-source is the name of an alternate database to connect to. If not present, it defaults to "LocalServer".

When the Thoroughbred client and the database server reside on the same physical system, you can specify a loop-back address.

TCP/IP-address is the TCP/IP address of the server.

When the Thoroughbred client and the database server reside on the same physical system, you can specify a loop-back address.

: separates the *host-name*|*TCP/IP-address* from the *TCP/IP-port*.

TCP/IP-port is the TCP/IP port number. The default is **5673**.

Specify a value to avoid conflict with another TCP/IP process that requires the default port number. The specified port number will be used by the transport protocol layer of TCP/IP to deliver the packet data to the requested application.

Examples of entries in the SERVER.MAP file follow:

```
OR:server1:5673
OR:198.189.167.20:5673
```

In the first line, the *server-ID* is **OR**, the *host-name* is **server1**, and the *TCP/IP-port* is **5673**. The default has been specified as the value of the *TCP/IP-port* parameter.

In the second line, the *server-ID* is **OR**, the *TCP/IP-address* is **198.189.167.20**, and the *TCP/IP-port* is **5673**. The default has been specified as the value of the *TCP/IP-port* parameter.

Note: Comments beginning with # in position 1 are allowed in SERVER.MAP file. Lines that do not contain a colon are considered comments.

For more information on the SERVER.MAP file, please refer to TS Oracle DataServer documentation.

7.2 TS Network DataServer

The SERVER.MAP file helps to enable new or existing applications to share data across a network. It establishes a relationship between a two-character server ID used by Thoroughbred applications and the server system.

If the file does not exist, you can create it as a simple ASCII file. If SERVER.MAP is not located in the current directory, the system searches:

- **/usr/lib/basic** directory, for UNIX/Linux, or
- **C:\WINDOWS** for Windows or **C:\WINNT** for Windows NT.

Entries in the SERVER.MAP file are formatted in the following way:

*server-ID***:**{*host-name*|*TCP/IP-address*}[**:***TCP/IP-port*][*:PARENT*][*:SECURE*]

server-ID is a two-character ID that refers to a server.

: separates the *server-ID* from the *host-name*.

host-name is the host name of the server system.

TCP/IP-address is the TCP/IP address of the server.

: separates the *host-name*|*TCP/IP-address* from the *TCP/IP-port*.

TCP/IP-port (optional) is the TCP/IP port number. The default is **5680**.

Specify a value to avoid conflict with another TCP/IP process that requires the default port number. The specified port number will be used by the transport protocol layer of TCP/IP to deliver the packet data to the requested application.

: separates the *TCP/IP-port* from optional keywords.

PARENT (optional) is a keyword designating a server as the host for TS Network Basic.

SECURE (optional) is a keyword specifying that secure communications with the server are required.

Examples of entries in the SERVER.MAP file follow:

```
S1:server1:5680
S2:198.189.167.20:5680
S3:mainserver::SECURE
```

In the first line, the *server-ID* is **S1**, the *host-name* is **server1**, and the *TCP/IP-port* is **5680**. The default has been specified as the value of the *TCP/IP-port* parameter.

In the second line, the *server-ID* is **S2**, the *TCP/IP-address* is **198.189.167.20**, and the *TCP/IP-port* is **5680**. The default has been specified as the value of the *TCP/IP-port* parameter.

In the third line, the *server-ID* is **S3**, the *host-name* is **mainserver**. This is a secure connection using the default *TCP/IP-port*.

Note: Comments beginning with # in position 1 are allowed in SERVER.MAP file. Lines that do not contain a colon are considered comments.

For more information on the SERVER.MAP file, please refer to TS Network DataServer documentation.

7.3 Thoroughbred Basic ODBC Client Capability

The SERVER.MAP file helps establish the interface and connection between Thoroughbred products and ODBC-compliant databases. It is located in the Thoroughbred root directory. If the file does not exist, you can create it as a simple ASCII file.

Entries in the SERVER.MAP file are formatted in the following way:

*server-ID***:***data-source*

server-ID is a two-character ID that refers to a server.

: separates the *server-ID* from the *data-source*.

data-source is the data source name specified to Microsoft Windows.

Valid data source names can contain the letters **A** through **Z**, the letters **a** through **z**, and the digits **0** through **9**. Data source names are case-sensitive.

Assuming that *server-ID* has been specified as **S1** and *data-source* as **NewSource**, you can specify the following valid entry in the SERVER.MAP file:

```
S1:NewSource
```

Note: Comments beginning with # in position 1 are allowed in SERVER.MAP file. Lines that do not contain a colon are considered comments.

For more information on the SERVER.MAP file, please refer to the Thoroughbred Basic ODBC Client Capability Customization Supplement.

8. The IPLINPUT File

This operating system file defines attributes Thoroughbred Basic uses to function within the operating system environment. The IPLINPUT file is an initial program load (IPL) file. By default, it is loaded into memory whenever a user starts Thoroughbred Basic. You can use the information in the following subsections to modify the IPLINPUT file or to create customized IPL files.

The IPLINPUT file is the most complex system file Thoroughbred Basic uses. It specifies:

- The amount of memory to allocate to each task.
- How error handling is controlled.
- The number of files that can be used simultaneously for each task.
- What devices and ports are available to each task, how each task is to be initialized, and so on.

Section 8.1 contains examples of IPLINPUT files. The sections that follow 8.1 contain descriptions of IPLINPUT file components.

8.1 IPLINPUT file examples

This example displays the contents of a small IPLINPUT file:

```
CNF 1,3,1,18
PTN 1,90000
DEV D0,1,,,,,,UTILS
DEV T0,7,,,,,,tty
DEV LP,4,,,,,,lp
IPL 1,2,T0
END
```

The following example contains the contents of a much larger IPLINPUT file:

```
CNF 1,11,1,60,CUTERR,0,0,D,.

PTN 1,120000
```

PRM ALLOC
PRM BASE-YEAR=year
PRM BTRIEVE
PRM CGA
PRM CMASK
PRM COBOL
PRM CREATWDBATTR
PRM CVTSTRIP
PRM DEBUG=progname
PRM DISABLE
PRM DONTCHECKTEXT
PRM EDIT=EDITPROG
PRM EDITF=EDITPROG
PRM EDITPUBLICS
PRM ERRMASK
PRM FULL-COMPARE
PRM IEEESWAP
PRM IF47
PRM IPCKEY=23
PRM LISTPAREN
PRM LOCKBYCHANNEL
PRM LONG-PROMPT
PRM LONGVAR
PRM NOBANNER
PRM NOROUND
PRM NOTRANS
PRM NOWINCLOSE
PRM OFF-ERR127
PRM OPENLIB=file-channel
PRM Oracle-LOGIN=login/password
PRM ORA_DONTWRITENULLS
PRM ORA_NVLNULLS
PRM PGCHARBASE =C
PRM PUBLICS =nnn
PRM QUIT=3
PRM READONLY
PRM RMS sd,sm,ss
PRM SENTINEL
PRM SEP=8B
PRM SERIAL-EOF
PRM SHORT-ERROR
PRM SHORTIF
PRM SLEEPLOCK
PRM SMC create, search
PRM SMPLOCK
PRM SOCKETPOLLRATE=Numeric
PRM STACK=16384
PRM SYSMEM=8192
PRM SYSTEM=/usr/prevent/entry
PRM UNIQUE-KEYS

```
PRM UPPER
PRM VAR-NOTSET-ERR
PRM WAITLOCK=SECONDS
PRM WINDOWCOUNT=35

DEV D0,1,,,0,,,UTILS
DEV D1,1,,,1,,,IDL4
DEV D7,1,,,,,,D:\COMPANY\ACTG
DEV D8,1,,,,1,,
DEV G0,8
DEV G1,8
DEV T0,7,,,,,,tty
DEV T1,7,,,,,,tty05b
DEV LP,4,,142,1,1,,lp -dP4
DEV P1,4,,132,,2,5,tty
DEV P2,4,,,,,,ttp8e
IPL 1,2,T0,**PSD
END
```

The IPLINPUT file has a rigid format. The different types of statements must appear in the order displayed above. The sample statements displayed in the above example are fully described in the following sections:

CNF statements specify configuration information. This statement is described in Section 8.2.

PTN statements specify partition information. This statement is described in Section 8.3.

PRM statements specify Thoroughbred Basic environmental parameters. These statements are described in Section 8.4.

DEV statements specify logical and physical devices Thoroughbred Basic can use. These statements are described in Section 8.5.

IPL statements specify the program Thoroughbred Basic loads when it is started. This statement is described in Section 8.6.

END statements specify the end of the IPLINPUT or IPL file. This statement is described in Section 8.7.

Inclusion of operating system environment variables, which is not illustrated in the above examples, is discussed in Section 8.8.

The IPLPRM file contains environment parameters that apply to all users of Thoroughbred Basic. For more information on the IPLPRM file please refer to Chapter 9.

8.2 The CNF statement

The **CNF** statement specifies configuration information. An IPL file contains only one **CNF** statement, which must occupy the first line of the file.

8.2.1 Overview of the CNF statement

The **CNF** statement has the following format:

CNF *numparts*,*numdevs*,*numtasks*,*numofiles*,*progerr*,*traceflag*,*ofilesflag*,*datefrmt*,*dfrmtsep*, *strtJulday*,*endJulday*

CNF begins the **CNF** statement.

numparts is the number of partitions. The only valid value is **1**. You must specify the value for this parameter.

numdevs is the number of device definition lines that follow. You must specify a value for this parameter. For more information on device definition lines, please refer to the subsection on **The DEV statement**.

numtasks is the number of tasks in this partition. The only valid value is **1**. You must specify the value for this parameter.

numofiles is the number of available open file table entries. Valid values cannot exceed two less than the maximum number of files per task, which is specified by the operating system. You must specify a value for this parameter.

Each open file and each program in memory uses one file table entry. Thoroughbred Basic public programs in memory do not use file table entries.

progerr is the program to run to manage automatic escape and error processing. A valid value is the name of such a program. The default is no name, which means no automatic processing.

You can specify the **CUTERR** program, which is provided with Thoroughbred Basic. This program prevents escapes or errors from placing the user in Thoroughbred Basic Console Mode.

If the program name is preceded by a **.** (period) Thoroughbred Basic will **CALL** the program rather than **RUN** the program. For more information on the **CALL** and **RUN** directives please refer to Volumes I and III of the Thoroughbred Basic Language Reference.

For more information on escape and error processing, please refer to the chapter on **Program Control** in the Thoroughbred Basic Developer Guide. For more information on Thoroughbred Basic Console Mode, please refer to the Thoroughbred Basic Developer Guide.

traceflag is the trace/console mode flag for automatic escape and error processing. This parameter is ignored unless an automatic escape and error processing routine is specified for the *progerr* parameter. Valid values are:

0 allows for "T" and "C" responses. This is the default.

1 disallows "T" and "C" responses.

ofilesflag Enables or disables the open files caching and file sharing features. Valid values are:

0 open files caching is disabled; file sharing is enabled. This is the default.

1 open files caching is disabled; file sharing is enabled.

2 open files caching is enabled; file sharing is enabled.

3 open files caching is enabled; file sharing is disabled.

4 open files caching is disabled; file sharing is disabled.

When enabled, the open files caching feature allows files to remain opened by the operating system even though a Thoroughbred Basic program has issued a CLOSE directive. This can save time spent closing and opening files in complex file systems but increases the load on operating system resources.

The file-sharing feature is only for Microsoft Windows operating systems. When enabled, extra steps are taken to insure that files can be shared with other systems using mapped drives. This is accomplished by flushing the local data cache after each update. Disabling this feature when mapped drives are not used will improve performance.

This setting can be overridden on individual DEV statements. The settings are displayed using the DUMP IPLDEVS "OPTIONS=Y" directive.

datefrmt is the date format. Valid values are:

M for *MMDDYY*. This is the default.

D for *DDMMYY*.

Y for *YYMMDD*.

MM is short for month, *DD* is short for day, and *YY* is short for year.

dfrmtsep is the date format separator, which is used to separate the *MM*, *DD*, and *YY* components of the *datefrmt* clause. Valid values are any character except for the null character. The **/** (slash) character is the default.

strtJulday is the starting Julian day, which you can use to set daylight savings time. Thoroughbred Basic recognizes the daylight savings time parameters specified in the TZ environment variable. If you prefer, you can set daylight savings time with the *strtJulday* and *endJulday* parameters. This parameter has the following format:

Julianday[**:***hour*]

Julianday is an integer that specifies the Julian day.

hour is an integer that specifies the hour of the day. The default is **2**.

endJulday is the ending Julian day, which you can use to set daylight savings time. This parameter has the following format:

Julianday[**:***hour*]

Julianday is an integer that specifies the Julian day.

hour is an integer that specifies the hour of the day. The default is **2**.

Note: Allows daylight savings time arguments on CNF records to be reversed for countries south of the Equator, where the end date is before the start date.

8.2.2 Examples of the CNF statement

CNF 1,**11**,**1**,**60**,**CUTERR**,**0**,**0**,**D**,**.**

This example specifies *DDMMYY* as the date format and **.** (period) as the character that separates date components. For example, January 30, 1995 will be formatted as **30.01.95**.

CNF 1,**11**,**1**,**60**,**CUTERR**,**0**,**0**,,,**90**:**3**,**300**

This example sets the start of daylight savings time to the 90th Julian day at 3:00 A.M. and the end of daylight savings time to the 300th Julian day at 2:00 A.M.

8.3 The PTN statement

The **PTN** statement specifies the memory partition configuration for the task. An IPL file contains only one **PTN** statement, which must occupy the second line of the file.

8.3.1 Overview of the PTN statement

The **PTN** statement has the following format:

PTN *partnum*,*partsize*

PTN begins the **PTN** statement.

partnum is the partition number. The only valid value is **1** You must specify the value for this parameter.

partsize is the number of bytes of memory in the partition. This space is only for data. Program space is allocated from the remaining available system memory.

8.3.2 Example of the PTN statement

PTN 1,12000

8.4 The PRM statement

PRM statements are optional statements that define Thoroughbred Basic environment parameters. These statements tell Thoroughbred Basic how to interpret certain situations and how to change its interface with the operating system. Beginning with Thoroughbred Basic version 8.5.0 PRM statements can also be placed in the IPLPRM global parameter file.

Note: The IPLPRM can be disabled by setting the IPLPRM variable to **N** on individual systems. Example:
IPLPRM=N

8.4.1 Overview of the PRM statement

PRM statements must appear after the **PTN** statement and before any **DEV** statements. There is no limit to the number of **PRM** lines you can specify.

You can place multiple **PRM** statements on the same line, but you must use a comma to separate one statement from the next. Starting with Thoroughbred Basic 8.2, **PRM** statements that contain hyphenation can use either a dash or an underscore. For example: **PRM LONG-PROMPT** can also be specified as **PRM LONG_PROMPT**.

The **PRM** statement has the following format:

PRM *clause1*[[**,PRM** *clause2*] **. . .**]

Valid values for **PRM** *clauses* are listed in the following section.

8.4.2 PRM statement list

ALLOC

Used by some file creation directives to reallocate physical disk storage. It causes the physical file to be allocated and erased to null when a **CREATE** directive is issued. Since UNIX allocates disk file space as it is used, this parameter ensures that physical space is available for the key area when the file is created.

Running out of space when **WRITE**ing a data record generates an **ERR=02** and may result in the loss of a data record. Running out of space in a keyblock area could cause the loss of key pointers.

BASE-YEAR=*year*

Specifies a base year for 00, a two-digit date. In most cases, the value for *year* is **1800** or **1900**. The default is **1900**.

This PRM statement can help you circumvent problems associated with century rollover, specifically the transition from 1999 to 2000. For more information on **PRM BASE-YEAR** please refer to Thoroughbred's YEAR 2000 White Paper.

BTRIEVE

On systems that support the BTRIEVE format, **PRM BTRIEVE** enables Thoroughbred Basic to create files in BTRIEVE format. This statement does not affect files that have already been saved in another format.

PRM BTRIEVE and **PRM COBOL** are mutually exclusive. If both **PRM** statements are specified, or if BTRIEVE is not active, Thoroughbred Basic will generate an error.

CGA

Invokes a special color graphics driver to handle older color graphics adapters. Use this clause only if your adapter is producing "snow" on the monitor. The **PRM CGA** statement may result in reduced input/output performance.

PRM CGA is only available under MS-DOS.

CMASK

Allows you to set the default values for CMASK for the entire session.

PRM CMASK[=|]"*cmask_arg*"

cmask_arg is a two part field and must be enclosed in double quotes. The first character must be either a decimal point (.) or a comma (,) indicating the default to be used in numeric values. The remainder of the string is the value to be substituted for dollar signs. It can be from 1 to 8 characters in length. This PRM can be used with the TS ODBC DataServer and also in Basic. See SET CMASK in the Basic Language Reference (R-Z).

IPLINPUT	**PROGRAM**
PRM CMASK ","	SET CMASK ".=,"
PRM CMASK ".YEN"	SET CMASK "$=YEN"
PRM CMASK "."	SET CMASK ".=.|$=$"
PRM CMASK ",YEN"	SET CMASK ".=,|$=YEN"

COBOL

Specifies that **MSORT** files will be created in Micro Focus COBOL format. Thoroughbred Basic can read and write these files.

PRM COBOL and **PRM BTRIEVE** are mutually exclusive. If both **PRM** statements are specified, Thoroughbred Basic will generate an error.

For more information on the **MSORT** directive, please refer to the Thoroughbred Basic Language Reference.

CREATWDBATTR

Helps establish the interface between Thoroughbred products and Oracle databases. It specifies that new tables will be created using the attributes from the system dictionary.

For more information on the **PRM CREATWDBATTR** statement, please refer to TS Oracle DataServer documentation.

CVTSTRIP

Removed the unconditional masking to 7-bit characters by the CVT ( ) function. This can be stored by using PRM CVTSTRIP.

DEBUG=*progname*

This **PRM** statement defines a program you can use to debug Thoroughbred Basic applications.

If this statement is specified and you use the **Ctrl-B** keystroke to interrupt program execution, *progname* will execute. When a debugging program exists, the interrupted Thoroughbred Basic program will continue to execute. The value of the **CTL** system variable is set to -99,999.

If this statement is not specified and you use the **Ctrl-B** keystroke to interrupt program execution, Thoroughbred Basic will display the following message:

```
Return to UNIX (Y/N)?
```

Type either **Y** for yes or **N** for no and press the **Enter** key.

For more information on the **CTL** system variable, please refer to the Thoroughbred Basic Language Reference.

DISABLE

Specifies that a logical directory can be disabled by the **DISABLE,LOCAL** directive even if the directory contains open files.

If **PRM DISABLE** is not specified, the attempt to disable a directory that contains open files generates ERROR 0.

For more information on the **DISABLE** directive and its **LOCAL** parameter, or on the **OPEN** directive, please refer to the Thoroughbred Basic Language Reference.

DONTCHECKTEXT

This **PRM** statement may be useful to developers who use special characters that cannot be printed in English. Printable characters range from **20** to **$7E$** and the Thoroughbred portable graphics characters range from **$C0$** to **CF**.

The **WINDOW PUT** directive maps unprintable characters onto the * (asterisk), which is a printable character. If **PRM DONTCHECKTEXT** is specified, the **WINDOW PUT** directive does not map unprintable characters onto the *****. The **PRM DONTCHECKTEXT** statement is available starting with Thoroughbred Basic 8.1.

DSTPRE2007

The begin and end dates for Daylight Saving Time were changed in 2007 and included in Basic version 8.6.0. This PRM will force the pre-2007 dates of the first Sunday in April through the last Sunday in October at 2 AM.

EDIT=*progname*

Specifies the name of the program editor used to edit Thoroughbred Basic programs. The **EDIT** command will run this program. The default is ***EDIT***.

For more information on ***EDIT***, please refer to the description of the **EDIT** full-screen directive in the Thoroughbred Basic Language Reference or the Thoroughbred Basic Utilities Manual.

EDITF=*progname*

Specifies the name of the program editor used to edit Thoroughbred Basic programs. The **EDITF** command will run this program. The default is **8EDITF**.

The **8EDITF** utility is available starting with Thoroughbred Basic 8.1.2. It is a full-screen program editor that enables you to select the level of formatting, use the editing keys, and read Thoroughbred Basic on-line documentation. In order to use **EDITF**, you must have installed Thoroughbred Dictionary-IV and set up your terminal for Thoroughbred Basic Windows For more information on **8EDITF**, please refer to the description of the **EDITF** full-screen directive in the Thoroughbred Basic Language Reference or the Thoroughbred Basic Utilities Manual.

EDITPUBLICS

When EDITPUBLICS IS "ON" an untrapped ESC key or an untrapped error will halt a public program at the current line in console mode. When EDITPUBLICS is "off" untrapped errors and the ESC key are reported as an error on the CALL statement.

For example: If you get an ERROR 42 on a CALL statement and you know the problem is in the Public Program, you can use EIDTPUBLICS. When you use this PRM while developing and testing, the Public Program will stop where the error occurred. You can then edit the program and save it.

ERRMASK

Generates an error whenever data length exceeds the length of a print mask.

Thoroughbred Basic tries to print data when a print mask is specified, even when the data exceeds the capacity of the mask. When this happens, Thoroughbred Basic does not generate a 40-series error. **PRM ERRMASK** causes Thoroughbred Basic to generate the error whenever the data exceeds the mask.

FULL-COMPARE

Specifies that all numeric comparisons in Thoroughbred Basic are made at full **FLOATING POINT**.

If this statement is not specified, and if **PRECISION** is set to **2**, then 1.005 tests equal to 1.006 because both values equal 1.01 when **PRECISION** is set to **2**. If **PRECISION** is set to **3** or higher, including **FLOATING POINT**, then the test returns a not-equal condition.

If **PRM FULL-COMPARE** is specified, the test returns a not equal regardless of the current setting of **PRECISION**. Besides a numeric comparison in an **IF/THEN/ELSE/FI** Directive, other situations which are sensitive to **PRM FULL-COMPARE** are the **FOR/NEXT** test for loop termination, **MIN/MAX** testing, **INPUT** of a numeric variable with verification, and **WHILE/WEND** loop control testing if the **WHILE** condition involves a numeric test.

For more information on **PRECISION** and **FLOATING POINT**, and on the directives and functions mentioned in the preceding paragraph, please refer to the Thoroughbred Basic Language Reference.

IEEESWAP

This statement changes the natural ordering of the bytes in a value defined using the IEEE standard to an ordering that is opposite.

The **STR** and **NUM** functions enable you to specify **NTP=8**, for IEEE single precision floating point, or **NTP=9**, for IEEE double precision floating point. The numbers are managed according to the natural byte ordering of the machine. **PRM IEEESWAP** enables you to **WRITE** files with IEEE numbers on one machine and **READ** the file on another machine that uses reverse byte ordering.

This parameter is available starting with Thoroughbred Basic 8.2.

IF47

Suppresses error 47 reporting within an **IF** statement. An error 47 causes the **IF** statement to be false rather than reporting the error. For example:

```
00100 LET A$ = "";
      IF A$(1,1) = "X" THEN
         PRINT "True"
      ELSE
         PRINT "False"
      FI
```

If **PRM IF47** is active the code above prints **False**. If **PRM IF47** is not active the above code reports an error 47 on line 00100.

For more information on error 47 please refer to the Thoroughbred Basic Developer Guide.

IPCKEY=*xx*

Enables you to specify a decimal Inter-Process Communication Key when Thoroughbred Basic must make shared memory facilities available to another product. Thoroughbred Basic uses a shared memory segment for ghost tasks.

The decimal number you can specify can be established only after you find the relevant information in the documentation for the operating system and the other product that will use shared memory.

This **PRM** statement is not available under MS-DOS.

LISTPAREN

This statement causes Thoroughbred Basic to print parentheses to enclose the dimension sizes for numeric arrays created by the **DIM** directive. The default is to use the bracket characters.

Prior to Thoroughbred Basic 8.0, parentheses were used to enclose dimension sizes of numeric arrays. Currently, parentheses are only used to enclose variable names used by the string version of the **DIM** directive. This convention makes it easier to distinguish the numeric array version of the directive from the string version of the directive.

The **PRM LISTPAREN** statement his has no impact on program execution.

LOADTRIGGER=

Besides LOAD, you can also activate a trigger using the IPLINPUT file.

```
PRM LOADTRIGGER=trigger-definition-name
```

There is no option for determining which Basic method will be used as the lead trigger program. When a trigger request is received by Basic, it invokes "OOIO". The necessary information about the request is passed to "OOIO" and it is then the responsibility of "OOIO" to analyze the data and determine what action is appropriate.

For more information on the **DIM** directive, please refer to the Thoroughbred Basic Language Reference.

LOCKBYCHANNEL

Restores the functionality of OPEN and LOCK commands that was available prior to Thoroughbred Basic 8.4.0. This **PRM** statement is available in Thoroughbred Basic 8.4.0 and above.

As default behavior, Thoroughbred Basic 8.4.0 and higher versions enable you to lock a channel to a file you have opened one or more times or open a file you have self-locked. The **LOCK** directive will prohibit other Thoroughbred Basic users from opening the file, but you can open the file. However, you cannot lock the file two or more times.

Previous versions of Thoroughbred Basic prohibited any user from opening a locked file, including the user who locked the file. To restore this functionality, use the **PRM LOCKBYCHANNEL** statement.

For more information on the **OPEN**, **LOCK**, and **UNLOCK** directives, please refer to the Thoroughbred Basic Language Reference.

LONG-PROMPT

Changes the screen prompt to display the name of the current program, a space character, the last generated Thoroughbred Basic error code, and the > character. The default is the > character.

The prompt specified by the **PRM LONG-PROMPT** statement may be helpful when you debug a Thoroughbred Basic program using single-stepping mode.

For more information on debugging, please refer to the chapter on **Program Control** in the Thoroughbred Basic Developer Guide.

LONGVAR

Sets the default syntax checker to long variable name mode, which enables variable names to be up to 33 characters long. Starting with Thoroughbred Basic 8.0, this is the default.

For more information on variable names, please refer to the descriptions of the **LONGVAR** and **SHORTVAR** directives in the Thoroughbred Basic Language Reference.

NOBANNER

Turns off the display of the Thoroughbred Basic banner that occurs when a user starts Thoroughbred Basic. The default is to allow Thoroughbred Basic to clear the screen and display the banner.

PRM NOBANNER is available starting with Thoroughbred Basic 8.2.

NOROUND

Specifies that numeric comparisons be made at full value. These comparisons are not subject to the current value set by the **PRECISION** directive. Numbers that need rounding are rounded at the end of the mathematical equation, not during the intermediate stages of numeric processing.

PRM NOROUND enables you to manage numeric comparisons in a way consistent with the method used in some other implementations of a business BASIC language.

For more information on the **PRECISION** directive, please refer to the Thoroughbred Basic Language Reference.

NOTRANS

Enables you to use transaction-processing directives without having to use any of the transaction processing facilities. This helps you produce portable code without affecting execution speed.

Program execution is affected in the following ways:

- The **LOG OPEN** directive can be executed. However, syntax and options are not checked. The log file and the Master Log are not opened.
- The **TRANSACTION BEGIN** directive simulates a **TRANSACTION BEGIN**, but no entries can be made to the log file. All records changed after the **TRANSACTION BEGIN** directive are locked, even though they cannot be entered into the log file.
- The **COMMIT** and **ROLLBACK** flush out the system buffers. All records that were locked become unlocked. Executing the **ROLLBACK directive** will generate an **ERR=13** because **ROLLBACK** cannot execute when **PRM NOTRANS** is set.
- The **LOG CLOSE** directive clears any outstanding internal flags.

For more information on transaction processing, please refer to the Thoroughbred Basic Developer Guide. For more information on the directives mentioned above, please refer to the Thoroughbred Basic Language Reference.

PRM NOTRANS is available starting with Thoroughbred Basic 8.2.

NOWINCLOSE

Prevents a user from exiting Thoroughbred Basic by pressing the **X** system box in the upper right-hand corner of the window.

PRM NOWINCLOSE is valid on systems running Thoroughbred Basic under Microsoft Windows. **PRM NOWINCLOSE** is available starting with Thoroughbred Basic.

OFF-ERR127

Specifies that Thoroughbred Basic does not set the **ERR** system variable to **127** when the **Escape** key is pressed. The default is to set **ERR** to **127**.

This statement enables you to press the **Escape** key and obtain the last generated error code contained in the **ERR** variable. This feature can help you debug Thoroughbred Basic programs.

For more information on the **ERR** system variable, please refer to the Thoroughbred Basic Language Reference.

PRM OFF-ERR127 is available starting with Thoroughbred Basic 8.1.

OPENLIB=*file*,*channel*

Opens the specified library on the specified channel. This statement causes Thoroughbred Basic to behave as if the following code were executed:

```
OPEN (channel, OPT="OLIB") filename
```

If the channel is missing or invalid, if the file is not found, or if the file is not a library, an error occurs and Thoroughbred Basic does not run.

This **PRM** statement is available starting with Thoroughbred Basic 8.2.

Oracle-LOGIN=*login*/*password*

Helps establish the interface between Thoroughbred products and Oracle databases. If multiple Oracle servers are configured, and the login and password are identical for each server, the **PRM Oracle-LOGIN** statement can replace multiple login and password specifications in **DEV** statements.

login is the Oracle login used to connect to the database.

/ separates *ID* from *password*.

password is the Oracle password used to connect to the database.

For more information on **DEV** statements, please refer to the subsection on the **DEV statement** in this chapter. For more information on the **PRM Oracle-LOGIN** statement, and on **DEV** statements relevant to TS Oracle DataServer, please refer to TS Oracle DataServer documentation.

ORA_DONTWRITENULLS

Helps establish the interface between Thoroughbred products and Oracle databases. The **PRM ORA_DONTWRITENULLS** statement specifies that null data is changed physically when the data is returned to a Thoroughbred Basic application.

The **PRM ORA_NVLNULLS** statement provides an alternate way of processing null data.

For more information on these **PRM** statements, please refer to TS Oracle DataServer documentation.

ORA_NVLNULLS

Helps establish the interface between Thoroughbred products and Oracle databases. The **PRM ORA_NVLNULLS** statement specifies that null data is changed logically when the data is returned to a Thoroughbred Basic application.

The **PRM ORA_DONTWRITENULLS** statement provides an alternate way of processing null data.

For more information on these **PRM** statements, please refer to TS Oracle DataServer documentation.

PGCHARBASE=*x*

Specifies that the business graphics characters are based on the hexadecimal character **$***x***0$**. Valid values for *x* are **8**, **9**, **A**, **B**, **C**, **D**, **E**, and **F**. The default is **C**.

Thoroughbred Basic uses 16 standard business graphics characters. Their hexadecimal values range from **$***x***0$** through **$***x***F$**. These characters are available starting with Thoroughbred Basic 8.1B2.

PUBLICS=*nnn*

Specifies the maximum number of Thoroughbred Basic public programs that can be loaded into memory. Valid values range from **5** through **200**. Under UNIX or Microsoft Windows, the default is **100**. Under MS-DOS the default is **20**.

The **ADDR** directive loads public programs into memory. The **DROP** and **DROP ALL** directives remove public programs from memory. For more information on public programs, please refer to the Thoroughbred Basic Developer Guide. For more information on the directives mentioned above, please refer to the Thoroughbred Basic Language Reference.

QUIT=*n*

Changes the control key sequence that halts Thoroughbred Basic to **Ctrl-***n*. Valid values are integers. The default is **2**, which enables the **Ctrl-b** keystroke sequence to halt Thoroughbred Basic.

For example, if you specify **PRM QUIT=3**, the **Ctrl-c** keystroke sequence will halt Thoroughbred Basic.

You can prevent users from accidentally halting a Thoroughbred Basic task by specifying **PRM QUIT=0**. In most cases, no key sequence can produce the 00 (null) character.

This statement is not available under MS-DOS.

READONLY

Specifies that Thoroughbred Basic programs can **READ** data records made available through an **EXTRACT** directive issued by another task. However, a **WRITE** directive cannot write data to the record until that data record is released.

PRM READONLY also enables the current task to **READ** a record that the same task made available through an **EXTRACT** directive that specified a different channel.

Previously, this feature was not available in single-user MS-DOS or on systems that use mandatory locking. Starting with release level 8.40, newly created data files will use a common locking mechanism so that systems that do not support advisory locking can use the **PRM READONLY** statement.

For more information on the directives mentioned above, please refer to the Thoroughbred Basic Language Reference.

RMS *sd*,*sm*,*ss*

On systems that support RMS formats, **PRM RMS** enables Thoroughbred Basic to create, read, or write files in RMS formats. To specify default RMS file structures, you can set the following parameters:

sd is the sector-number for **DIRECT** files. Valid values are **0** through **5**, which stand for sector-numbers 0 through -5. The default is **0**.

To override this value, specify a value for the *sector-num* parameter in the **DIRECT** directive.

sm is the sector-number for **MSORT** files. Valid values are **0, 1** and **7**, which stand for sector-numbers 0, -1 and –7. The default is **0**.

Specifying a 7 will create a TVMsort file, which is a special form of MSORT file that allows the use of field-separated key data fields in a high-speed RMS file.

To override this value, specify a value for the *sector-num* parameter in the **MSORT** directive.

ss is the sector-number for **SERIAL** files. Valid values are **0** through **6**, which stand for sector-numbers 0 through -6. The default is **0**.

To override this value, specify a value for the *sector-num* parameter in the **SERIAL** directive.

For more information on **PRM RMS**, RMS file structures, and how to use Thoroughbred Basic on systems that contain RMS, please refer to the on-line release notes. For more information on the Thoroughbred Basic directives mentioned above, please refer to the Thoroughbred Basic Language Reference.

If **PRM RMS** is not specified, Thoroughbred Basic will assume that files are managed using Thoroughbred formats.

SENTINEL

Specifies that Thoroughbred Basic use the standard method to control access to DIRECT and SORT files.

On some UNIX systems with Symmetric Multiprocessing (SMP) **PRM SMPLOCK** may be the default. This PRM statement specifies an alternate method to control access to DIRECT and SORT files.

To override **PRM SMPLOCK**, specify **PRM SENTINEL**.

SEP=*nn*

Sets the value of the field separator character, which is contained in the **SEP** variable. Valid values are two hexadecimal characters. The default is **8A**.

If an invalid value is specified, Thoroughbred Basic will display an error message and it will not execute the command.

For more information on the **SEP** variable, please refer to the Thoroughbred Basic Language Reference.

This statement is available starting with Thoroughbred Basic 8.2.

SERIAL-EOF

Disables the automatic expansion of **SERIAL** files. If Thoroughbred Basic reaches the end of a file while using the **WRITE** directive to add data to a **SERIAL** file, it will generate an **ERR=02**.

End of file conditions can occur when a program tries to add more than the number of records specified for the file. It can also occur when a program tries to add a record that is larger than the amount of available space in the file. The total amount of space specified for the file is determined by the number of records multiplied by the average size of a record.

For more information on **SERIAL** files and the **WRITE** directive, please refer to the Thoroughbred Basic Language Reference.

PRM SERIAL-EOF is available starting with Thoroughbred Basic 8.2.

SHORT-ERROR

Specifies how errors are displayed. If this statement is in effect, and an untrapped error causes Thoroughbred Basic to go to Thoroughbred Basic Console Mode, only the portion of the code where the error occurred will be displayed. The default is to display the entire statement.

For more information on Thoroughbred Basic Console Mode, and on how Thoroughbred Basic processes errors, please refer to the Thoroughbred Basic Developer Guide.

PRM SHORT-ERROR is available starting with Thoroughbred Basic 8.2.

SHORTVAR

Sets the default syntax checker to short variable name mode, which enables variable names to be up to 2 characters long. The default is long variable name mode, which enables variable names to be up to 33 characters long.

When **SHORTVAR** is active, all Thoroughbred Basic Console Mode statements, and all **EXECUTE** and **MERGE** directives issued through Thoroughbred Basic Run Mode, are checked to ensure that only short variable names are used. **SHORTVAR** only allows the directives, functions, and variables that existed prior to Thoroughbred Basic 8.0. It does not affect execution of existing programs that contain Thoroughbred Basic 8.0 features and long variable names.

The **LONGVAR** directive or **PRM LONGVAR** enable you to use short or long variable names and make it easier to use new Thoroughbred Basic features. **LONGVAR** is the default.

For more information on variable names, please refer to the descriptions of the **LONGVAR** and **SHORTVAR** directives in the Thoroughbred Basic Language Reference. Please refer to the description of **PRM LONGVAR** in this subsection.

PRM SHORTVAR is available starting with Thoroughbred Basic 8.0.

SLEEPLOCK

Specifies that Thoroughbred Basic can use an alternate method to gain access to a locked record.

When Thoroughbred Basic issues a test lock UNIX system call and the request cannot be satisfied immediately, the call returns. Under this alternate method, an unsatisfied request causes Thoroughbred Basic to suspend execution for one second, then try again. If the lock request is unsatisfied after 15 attempts, an **ERR=0** is returned to the Thoroughbred Basic program.

PRM SLEEPLOCK is useful on a multi-processor system where many users must gain access to the same set of files. However, this alternate method may decrease system performance.

SMC *create,search*

Provides support for previous SMC Basics on DEC VAX machines. It enables you to specify whether the **.SMC** extension is appended to file names when you create or search for files.

To define when the extension is appended, specify the following parameters:

create specifies whether the **.SMC** extension is added when Thoroughbred Basic creates a file. Valid values are:

0 means that the extension is never appended.

1 means that the extension is always appended.

search specifies whether the **.SMC** extension is added when Thoroughbred Basic searches for a file. Valid values are:

0 means the extension is never appended. If the *create* parameter is **1**, do not specify this value for *search*.

1 means the extension is always appended. If the *create* parameter is **0**, do not specify this value for *search*.

2 means that Thoroughbred Basic will search for the file without adding the extension. If the file is not found, it will search for the file by adding the extension.

3 means that Thoroughbred Basic will search for the file by adding the extension. If the file is not found, it will search for the file with no extension.

PRM SMC has no effect on files that already have file extensions. For more information on **PRM SMC**, please refer to the on-line release notes.

SMPLOCK

Specifies that Thoroughbred Basic use an alternate method to control access to DIRECT and SORT files.

On some UNIX systems with Symmetric Multiprocessing (SMP) this method can speed the removal and addition of keys. The gains provided by this PRM statement can be determined by monitoring system performance.

PRM SMPLOCK must be added to the IPLINPUT files of all users.

For versions of Thoroughbred Basic that have **PRM SMPLOCK** as the default, **PRM SENTINEL** can be used to specify the standard method of access control.

SOCKETPOLLRATE

Only affects Telnet connections to the Windows Basic. The value supplied adjusts the rate that Basic checks the connection for an **ESC** or **CTL-B** entered by the user.

By default the connection is checked after every 1000th directive is executed. During development it may be desirable to use a lower number so that an **ESC** is recognized immediately. Performance can be improved significantly by using higher values, but with a corresponding loss of sensitivity to **ESC** and **CTL-B.**

STACK=*nnnnn*

Specifies the maximum number of bytes the Return Address Stack can use. Valid values are integers. The default is **8,192**.

The Return Address Stack contains information used to direct program flow and execution. If you use a great number of nested **GOSUB**, **WHILE/WEND**, or **FOR/NEXT** directives, you may need to increase the number of bytes in the stack.

For more information on the Return Address Stack, and on the directives mentioned above, please refer to the Thoroughbred Basic Language Reference.

This statement is available starting with Thoroughbred Basic 8.0.

SYSMEM=*nnnnn*

Specifies the number of bytes to set aside for the **SYSTEM** directive. Valid values are integers. The default is **4096**.

The number you specify must be large enough to contain the MS-DOS COMMAND.COM file.

For more information on the **SYSTEM** directive, please refer to the Thoroughbred Basic Language Reference.

This statement is only available under MS-DOS.

SYSTEM=*filename*

Causes the **SYSTEM** directive to execute a file. A valid value is the name of an existing file.

For example, you can prevent access to UNIX from a Thoroughbred Basic program by specifying a file such as **/usr/prevent/entry**.

For more information on the **SYSTEM** directive, please refer to the Thoroughbred Basic Language Reference.

This statement is not available under MS-DOS.

UNIQUE-KEYS

Used in the client Basic IPLINPUT file. Treats all secondary sorts unique by adding the primary sort to the end of all the secondary sorts when the various READ statements are translated to their equivalent SELECT statements on the server.

UPPER

Transforms all Thoroughbred Basic Console Mode keyboard input not enclosed by quotation marks into uppercase characters.

Thoroughbred Basic is case sensitive. All syntax requires uppercase. UNIX normally uses commands in lowercase. **PRM UPPER** can cause problems when a programmer must move between Thoroughbred Basic and the resident operating system.

PRM UPPER causes problems when you use the **EDIT** line directive to modify a program line that contains lowercase characters. This statement can cause problems when you refer to format or data names that contain lowercase letters.

For more information on the **EDIT** line directive, please refer to the Thoroughbred Basic Language Reference.

VAR-NOTSET-ERR

Specifies that a reference to a variable before a value has been assigned to the variable will generate **ERR=22**.

PRM VAR-NOTSET-ERR is available starting with Thoroughbred Basic 8.2.

WAITLOCK=*nn*

Specifies the amount of time that Thoroughbred Basic will wait for an extracted record to become available. The **READ**, **EXTRACT**, or other directives can extract a record from a file. The *nn* parameter specifies the number of seconds. Valid values range from **0** through **60**. The default is **15** seconds.

PRM READONLY overrides **PRM WAITLOCK**.

For more information on **PRM READONLY**, please refer to the description in this subsection. For more information on the **READ** or **EXTRACT** directive, please refer to the Thoroughbred Basic Language Reference.

This statement is available starting with Thoroughbred Basic 8.1.2.

WINDOWCOUNT=*nn*

Enables you to specify the maximum number of Thoroughbred Basic Windows. Valid values range from **5** through **200**. The default is **50**.

Because Thoroughbred Basic allocates memory for the number of Thoroughbred Basic Windows specified by this statement, you may want to save memory by specifying a smaller number.

For more information on Thoroughbred Basic Windows, please refer to the Thoroughbred Basic Developer Guide.

This statement is available starting with Thoroughbred Basic 8.1.

8.5 The DEV statement

The **DEV** statements tell Thoroughbred Basic what kinds of printers, terminals, directories, and tasks are available to Thoroughbred Basic applications and programs.

8.5.1 Overview of the DEV statement

DEV statements must appear after the **PRM** statements and before the **IPL** statement. If the IPL file does not contain **PRM** statements, **DEV** statements must be placed between the **PTN** statement and the **IPL** statement.

There must be at least one **DEV** statement to define the individual task and there must be at least one logical disk directory or communications port defined to provide for program and data access.

The **DEV** statement has the following generic format:

DEV *dev-ID*,*type*,*param-1*,*param-2*,*param-3*, *param-4*,*param-5*,*string*

The **DEV** statement is used to define the following types of devices to Thoroughbred Basic:

- Disk or directory devices
- ODBC-compliant server devices (Thoroughbred Basic ODBC Client Compatibility)
- Network server devices (TS Network DataServer)
- Oracle server devices (TS Oracle DataServer)
- SQL Server devices (TS DataServer for SQL Server)

  Terminal devices
- Printer devices
- Ghost tasks

Each type of **DEV** statement will be described in the following subsections.

8.5.2 DEV statement syntax for a disk or directory

DEV D*X*,*type*,*param-1*,*param-2*,*subdirflg*, *hierflg*,*param-5*,*dirname*

DEV begins the **DEV** statement. The statement begins in the leftmost column.

D*X* is the device ID, which Thoroughbred Basic uses to reference the disk or directory. Valid values range from **D0** through **D9** and **DA** through **DZ**. You can specify up to 36 logical and physical devices.

Disk or directory device specifications do not have to follow a set order. For example, you can specify the **D0** disk after you specify the **D1** disk.

type specifies the device type. For a disk or directory, the only valid value is **1**.

param-1 is a positional parameter. For disk or directory devices, do not specify a value for this parameter.

param-2 is a positional parameter. For disk or directory devices, do not specify a value for this parameter.

subdirflg is a positional parameter. For disk or directory devices, this parameter specifies that this directory can contain subdirectories and specifies the length of each subdirectory name. Valid values for this parameter are **1** through **9**. If you specify **1** for *subdirflg*, each subdirectory name must be three characters long, so specifying **1** is equivalent to specifying **3**.

Setting this attribute does not automatically create subdirectories. You must use operating system commands to create subdirectories.

Even if the defined length of subdirectory names is not three characters, you should create a **USR** subdirectory. If the first characters of a filename do not match any of the subdirectory names, the file will be placed in the **USR** subdirectory; if the **USR** subdirectory does not exist, Thoroughbred Basic will issue an error.

For more information on subdirectories, please refer to Section 4.4 of this manual.

hierflg is a positional parameter. For disk or directory devices, this parameter specifies whether Thoroughbred Basic will consider the disk a hierarchical directory. Valid values are:

0 The disk is not a hierarchical directory. This is the default.

1 The disk is a hierarchical directory.

A hierarchical directory specification enables Thoroughbred Basic to access all of the directories on the disk, and provides more flexibility in specifying directories and files. Directives such as **SET PREFIX** rely on the specification of a hierarchical directory. You can only specify one hierarchical directory in an IPL file.

For more information on the **SET PREFIX** directive, please refer to the Thoroughbred Basic Language Reference.

ofilesflag is a positional parameter. For disk or directory devices, this parameter enables or disables the open files caching and file sharing features. Valid values are:

0 The setting from the CNF statement is used. This is the default.

1 Open files caching is disabled; file sharing is enabled.

2 Open files caching is disabled; file sharing is enabled.

3 Open files caching is enabled; file sharing is disabled.

4 Open files caching is disabled; file sharing is disabled.

Use this parameter to override the *ofilesflag* setting on the CNF statement. This may be necessary for removable media such as CD drives. This parameter is automatically set to **1** for hierarchical directories. The open files caching and file sharing features are described in the CNF statement *ofilesflag* parameter.

dirname is a positional parameter. For disk or directory devices, this parameter specifies the name of the directory that will be used to locate files or subdirectories. The directory name does not have to be relative to the current working directory. The name can be absolute, relative, or logical.

Maximum *dirname* length is 64-characters. Longer specifications will be truncated, but no error message will be generated.

*Examples of the **DEV** statement for disk or directory devices*

Example: **DEV** statement for a logical disk directory

DEV D0,1,,,0,,,UTILS

DEV	Begins the **DEV** statement.
DEV **D0**	Specifies that the device is called **D0**.
DEV D0,**1**	Specifies the device type for **D0**. Valid values are: **1** logical disk directory **4** standard dot matrix, character printer, or laser printer **5** terminal that uses Thoroughbred Basic Windows **6** terminal that uses a video card **7** terminal that does not use Thoroughbred Basic Windows **8** ghost task **D0** is defined as a logical disk directory.
DEV D0,1,,,**0**	Specifies that there are no subdirectories in this directory. The default is **0**.
DEV D0,1,,,0,,,**UTILS**	Specifies that the system name for **D0** is **UTILS**. If **UTILS** does not exist, you must use operating system commands to create it. Thoroughbred Basic can refer to this device as **D0**, logical disk directory number **0** (based on the name **D0**), or by its name, **UTILS**.

Example: **DEV** statement for a logical disk directory with subdirectories

DEV D1,1,,,1,,,IDL4

DEV	Begins the **DEV** statement.
DEV **D1**	Specifies that the device is called **D1**.
DEV D1,**1**	Specifies the device type for **D1**. Valid values are:

1 logical disk directory
4 standard dot matrix, character printer, or laser printer
5 terminal that uses Thoroughbred Basic Windows
6 terminal that uses a video card
7 terminal that does not use Thoroughbred Basic Windows
8 ghost task

D1 is defined as a logical disk directory.

DEV D1,1,,,**1**

Specifies that this directory contains subdirectories. Placing files in subdirectories can increase Thoroughbred Basic execution speed by reducing the time required to find a file.

Valid values for the subdirectory flag are **1** through **9**. The flag tells Thoroughbred Basic that subdirectories exist and, with one exception, specifies the length of each subdirectory name. The exception is illustrated in the above example: if you specify **1** for the subdirectory flag, each subdirectory name must be three characters long, so specifying **1** is equivalent to specifying **3**.

In this example, each subdirectory must have a 3-character name. Each subdirectory will contain files with names that start with those three characters. For example, if the **WOR** subdirectory existed under **D1** on a UNIX system, a file named **WORKFILE** would be placed in **IDL4/WOR/WORKFILE**.

You should create a **USR** subdirectory. If the first characters of a filename do not match any of the subdirectory names, the file will be placed in the **USR** subdirectory.

Setting this attribute does not automatically create subdirectories. You must use operating system commands to create subdirectories.

For more information on subdirectories, please refer to Section 4.4 of this manual.

DEV D1,1,,,1,,,**IDL4**

Specifies that the system name for **D1** is **IDL4**. If **IDL4** does not exist, you must use operating system commands to create it.

Thoroughbred Basic can refer to this device as **D1**, logical disk directory number **1** (based on the name **D1**), or by its name, **IDL4**.

Example: **DEV** statement for an MS-DOS directory

DEV D7,1,,,,,,D:\COMPANY\ACTG

DEV

Begins the **DEV** statement.

DEV **D7**

Specifies that the device is called **D7**.

DEV D7,**1**	Specifies the device type for **D7**. Valid values are: **1** logical disk directory **4** standard dot matrix, character printer, or laser printer **5** terminal that uses Thoroughbred Basic Windows **6** terminal that uses a video card **7** terminal that does not use Thoroughbred Basic Windows **8** ghost task. Background tasks are not available to Thoroughbred Basic under MS-DOS. **D7** is defined as a logical disk directory.
DEV D7,1,,,,,,**D:\COMPANY\ACTG**	Specifies that the system name for **D7** is **D:\COMPANY\ACTG**. If this directory does not exist, you must use operating system commands to create it. Thoroughbred Basic can refer to this device as **D7**, logical disk directory number **7** (based on the name **D7**), or by its name, **D:\COMPANY\ACTG.**

Example: **DEV** statement for hierarchical disk directory

DEV D8,1,,,,1,,

DEV	Begins the **DEV** statement.
DEV **D8**	Specifies that the device is called **D8**.
DEV D8,**1**	Specifies the device type for **D8**. Valid values are: **1** logical disk directory **4** standard dot matrix, character printer, or laser printer **5** terminal that uses Thoroughbred Basic Windows **6** terminal that uses a video card **7** terminal that does not use Thoroughbred Basic Windows **8** ghost task **D8** is defined as a logical disk directory.
DEV D8,1,,,,,**1**,,	Specifies that Thoroughbred Basic will consider the disk a hierarchical directory. Valid values for this parameter are: **0** The disk is not a hierarchical directory. This is the default.

1 The disk is a hierarchical directory.

A hierarchical directory specification enables Thoroughbred Basic to access all of the directories on the disk, and provides more flexibility in specifying directories and files. Directives such as **SET DIR** and **SET PREFIX** rely on the specification of a hierarchical directory. You can only specify one hierarchical directory in an IPL file.

For more information on the directives mentioned above, please refer to the Thoroughbred Basic Language Reference.

8.5.3 DEV statement syntax for an ODBC server

DEV D*X*,*type*,*param-1*,*server-flag*,*param-3*,*param-4*,*param-5*,*server-ID*:*arguments*

DEV begins the **DEV** statement. The statement begins in the leftmost column.

D*X* is the device ID, which Thoroughbred Basic uses to reference the ODBC server. Valid values range from **D0** through **D9** and **DA** through **DZ**. You can specify up to 36 logical and physical devices.

ODBC server specifications do not have to follow a set order. For example, you can specify the **D0** disk after you specify the **D1** disk.

Note: The **D***X* specification means that ODBC servers are regarded as disk or directory devices. You can specify up to 36 disk or directory devices.

For more information on how ODBC servers work with Thoroughbred products, please refer to Chapter 5 of this manual.

type specifies the device type. For an ODBC server, the only valid value is **1**.

param-1 is a positional parameter. For ODBC servers, do not specify a value for this parameter.

server-flag is a positional parameter. It specifies that the disk directory is located on a server, and specifies the server type. For ODBC servers, the only valid value is **3**.

param-3 is a positional parameter. For ODBC servers, do not specify a value for this parameter.

param-4 is a positional parameter. For ODBC servers, do not specify a value for this parameter.

param-5 is a positional parameter. For ODBC servers, do not specify a value for this parameter.

server-ID[**:***arguments*] is a positional parameter. It specifies the ODBC, and any information needed to establish a connection to the ODBC-compliant database. Valid values are:

server-ID is the two-character specification assigned to the ODBC server. The *server-ID* specification in the **DEV** statement must match an entry in the **SERVER.MAP** file.

The **:***arguments* syntax element varies depending upon the *server-ID*. Some ODBC-compliant databases require only a valid *server-ID* specification.

: separates *server-ID* from the *arguments* specification.

arguments vary depending upon the *server-ID*. Some databases do not require *arguments*.

Some databases may require arguments that take the form of *login*/*password*:

login is the login used to connect to the database. The *login*/*password* specification is optional.

/ separates *ID* from *password*.

password is the password used to connect to the database. The *login*/*password* specification is optional.

For more information on *arguments*, please refer to the documentation for the relevant ODBC-compliant database management system.

Example: **DEV** statement for an ODBC server (Thoroughbred Basic ODBC Client Capability)

DEV D8,1,,3,,,,S1

DEV Begins the **DEV** statement.

DEV **D8** Specifies that the device is called **D8**.

DEV D8,**1** Specifies the device type for **D8**. Valid values are:

1 logical disk directory
4 standard dot matrix, character printer, or laser printer
5 terminal that uses Thoroughbred Basic Windows
6 terminal that uses a video card
7 terminal that does not use Thoroughbred Basic Windows
8 ghost task

D8 is defined as a logical disk directory.

DEV D8,1,,**3** Specifies that the disk directory is located on an ODBC server.

DEV D8,1,,3,,,,**S1** Specifies that the *server-ID* is **S1**. No further arguments are required.

Note: For more information on how to configure Thoroughbred products for use with ODBC servers and databases, please refer to the Thoroughbred Basic ODBC Client Capability Customization Supplement.

8.5.4 DEV statement syntax for a server (TS Network DataServer)

DEV D*X*,*type*,*param-1*,*server-flag*,*param-3*,*param-4*,*param-5*,*server-ID*:*path*

DEV begins the **DEV** statement. The statement begins in the leftmost column.

D*X* is the device ID, which Thoroughbred Basic uses to reference the server. Valid values range from **D0** through **D9** and **DA** through **DZ**. You can specify up to 36 logical and physical devices.

Network server specifications do not have to follow a set order. For example, you can specify the **D0** disk after you specify the **D1** disk.

Note: The **D***X* specification means that servers are regarded as disk or directory devices. You can specify up to 36 disk or directory devices.

For more information on how network servers work with Thoroughbred products, please refer to TS Network DataServer documentation.

type specifies the device type. For a server, the only valid value is **1**.

param-1 is a positional parameter. For network servers, do not specify a value for this parameter.

server-flag is a positional parameter. It specifies that the disk directory in located on a server, and specifies the server type. For Thoroughbred network servers, the only valid value is **2**.

param-3 is a positional parameter. For servers, do not specify a value for this parameter.

param-4 is a positional parameter. For servers, do not specify a value for this parameter.

param-5 is a positional parameter. For servers, do not specify a value for this parameter.

server-ID:*path* is a positional parameter. It specifies the server and the location of the server. Valid values are:

server-ID is the two-character specification assigned to the server. The *server-ID* specification in the **DEV** statement must match an entry in the SERVER.MAP file.

: separates server-ID from the *path* specification.

path	is the directory on the server that contains required data. The *path* specification can be fully qualified. If it is not fully qualified, the path specification is interpreted as relative to the directory that contains the server executable.

*Examples of the **DEV** statement for server devices (TS Network DataServer)*

Example: **DEV** statement with a fully qualified *path* specification

DEV D1,1,,2,,,,S1:/home/data/user

DEV	Begins the **DEV** statement.
DEV **D1**	Specifies that the device is called **D1**.
DEV D1,**1**	Specifies the device type for **D1**. Valid values are: **1** logical disk directory **4** standard dot matrix, character printer, or laser printer **5** terminal that uses Thoroughbred Basic Windows **6** terminal that uses a video card **7** terminal that does not use Thoroughbred Basic Windows **8** ghost task **D1** is defined as a logical disk directory.
DEV D1,1,,**2**	Specifies that the disk directory is located on a network server.
DEV D1,1,,2,,,,**S1:/home/data/user**	Specifies that the *server-ID* is **S1**, and the *path* is **/home/data/user**. The files on the **D1** device are located in the **/home/data/user** directory on the **S1** server.

Example: **DEV** statement with a relative *path* specification

DEV D2,1,,2,,,,S2:data/user

DEV	Begins the **DEV** statement.
DEV **D2**	Specifies that the device is called **D2**.
DEV D2,**1**	Specifies the device type for **D2**. Valid values are: **1** logical disk directory

4 standard dot matrix, character printer, or laser printer

5 terminal that uses Thoroughbred Basic Windows

6 terminal that uses a video card

7 terminal that does not use Thoroughbred Basic Windows

8 ghost task

D2 is defined as a logical disk directory.

DEV D2,1,,**2** — Specifies that the disk directory is located on a network server.

DEV D2,1,,2,,,,**S2:data/user** — Specifies that the *server-ID* is **S2**, and the *path* is **data/user**, a relative path specification. For example, if the server executable is located in the **/usr/lib/basic** directory, the files on the **D2** device are located in the **usr/lib/basic/data/user** directory on the **S2** server.

Note: For more information on how to configure Thoroughbred products for use with network servers, please refer to TS Network DataServer documentation.

8.5.5 DEV statement syntax for an Oracle server (TS Oracle DataServer)

DEV D*X*,*type*,*param-1*,*server-flag*,*param-3*,*logon-cache*,*param-5*,*server-ID*:*login*/*password*

DEV begins the **DEV** statement. The statement begins in the leftmost column.

D*X* is the device ID, which Thoroughbred Basic uses to reference the Oracle server. Valid values range from **D0** through **D9** and **DA** through **DZ**. You can specify up to 36 logical and physical devices.

Oracle server specifications do not have to follow a set order. For example, you can specify the **D0** disk after you specify the **D1** disk.

Note: The **D***X* specification means that Oracle servers are regarded as disk or directory devices. You can specify up to 36 disk or directory devices.

For more information on how Oracle servers work with Thoroughbred products, please refer to TS Oracle DataServer documentation.

type specifies the device type. For an Oracle server, the only valid value is **4**.

param-1 is a positional parameter. For Oracle servers, do not specify a value for this parameter.

server-flag is a positional parameter. It specifies that the disk directory is located on a server, and specifies the server type. For Oracle servers, the only valid value is **4**.

param-3 is a positional parameter. For Oracle servers, do not specify a value for this parameter.

logon-cache is a positional parameter. It specifies the maximum number of concurrent active logins to the Oracle database. Valid values are numeric. This parameter is optional.

param-5 is a positional parameter. For Oracle servers, do not specify a value for this parameter.

server-ID:*login*/*password* is a positional parameter. It specifies the Oracle server, and the login and password used to enter the database. Valid values are:

server-ID is the two-character specification assigned to the Oracle server. The *server-ID* specification in the **DEV** statement must match an entry in the SERVER.MAP file.

: separates server-ID from the *login*/*password* specification.

login is the Oracle login used to connect to the database. The *login*/*password* specification is optional.

/ separates *ID* from *password*.

password is the Oracle password used to connect to the database. The *login*/*password* specification is optional.

Note: If multiple Oracle servers are configured, and the login and password are identical for each server, the **PRM Oracle-LOGIN** statement can replace multiple login and password specifications in **DEV** statements.

For more information on the **PRM Oracle-LOGIN** statement, on the SERVER.MAP file, and on **DEV** statements relevant to TS Oracle DataServer, please refer to TS Oracle DataServer documentation.

*Example of the **DEV** statement for Oracle server devices (TS Oracle DataServer)*

Example: **DEV** statement for an Oracle server

DEV D8,1,,1,,,,O1:jill/jij

DEV Begins the **DEV** statement.

DEV **D8** Specifies that the device is called **D8**.

DEV D8,**1** Specifies the device type for **D8**. Valid values are:

1 logical disk directory
4 standard dot matrix, character printer, or laser printer
5 terminal that uses Thoroughbred Basic Windows
6 terminal that uses a video card
7 terminal that does not use Thoroughbred Basic Windows
8 ghost task

D8 is defined as a logical disk directory.

DEV D8,1,,**1** Specifies that the disk directory is located on an Oracle server.

DEV D8,1,,1,,,,**O1:jill/jij** Specifies that the *server-ID* is **O1**, the *login* is **jill**, and the *password* is **jij**.

Note: For more information on how to configure Thoroughbred products for use with Oracle servers and databases, please refer to TS Oracle DataServer documentation.

8.5.6 DEV statement syntax for a SQL Server (TS DataServer for SQL Server)

DEV D*X***,***type***,***paran-1,server-flag,parm-3, logon-cache,parm-5,server-ID***:***login//password*

DEV begins the **DEV** statement. The statement begins in the leftmost column.

D*X* is the device ID, which Thoroughbred Basic uses to reference the SQL Server. Valid values range from **D0** through **D9** and **DA** through **DZ**. You can specify up to 36 logical and physical devices.

SQL Server specifications do not have to follow a set order. For example, you can specify the D0 disk after you specify the D1 disk.

Note: The **D***X* specification means that SQL Servers are regarded as disk or directory devices. You can specify up to 36 disk or directory devices.

For more information on how SQL Servers work with the Thoroughbred products, please refer to the TS DATASERVER for SQL Server documentation.

type specifies the device type. For SQL Server, the only valid value is **1**.

param-1 is a positional parameter. For SQL Servers, do not specify a value for this parameter.

server-flag is a positional parameter. It specifies that the disk directory is located on a server, and specifies the server type. For SQL Servers, the only valid value is **4**.

parm-3 is a positional parameter. For SQL Servers, do not specify a value for this parameter.

logon-cache is a positional parameter. It specifies the maximum number of concurrent active logins to the database. Valid values are numeric. This parameter is optional.

param-5 is a positional parameter. For SQL Servers, do not specify a value for this parameter.

server-ID:*login/password* is a positional parameter. It specifies the SQL Server, and the login and password used to enter the database. Valid values are:

server-ID is the two-character specification assigned to the SQL Server. The *server-ID* specification in the DEV statement must match an entry in the SERVER.MAP file.

: separates server-ID from the *login/password* specification.

login is the SQL Server login used to connect to the database. The *login/password* specification is optional.

/ separates *ID* from *password.*

password is the SQL Server password used to connect to the database. The *login/password* specification is optional.

*Example of the **DEV** statement for SQL Server devices (TS DataServer for SQL Server)*

Example: **DEV** statement for a SQL Server

DEV D8,1,,1,,,,O1:jill/jij

DEV Begins the **DEV** statement.

DEV **D8** Specifies that the device is called **D8**.

DEV D8,**1** Specifies the device type for **D8**. Valid values are:

1 logical disk directory
4 standard dot matrix, character printer, or laser printer
5 terminal that uses Thoroughbred Basic Windows
6 terminal that uses a video card
7 terminal that does not use Thoroughbred Basic Windows
8 ghost task

D8 is defined as a logical disk directory.

DEV D8,1,,**4** Specifies that the disk directory is located on SQL Server.

DEV D8,1,,4,,,,**O1:jill/jij** Specifies that the *server-ID* is **O1**, the *login* is **jill**, and the *password* is **jij**.

Note: For more information on how to configure Thoroughbred products for use with SQL Servers and databases, please refer to the TS DataServer for SQL Server documentation.

8.5.7 DEV statement syntax for a terminal

DEV T*X*,*type*,*baud*,*parity*,*width*,*stopbits*,*nullstrip*,*driver*\|*port*

DEV begins the **DEV** statement. The statement begins in the leftmost column.

T*X* is the device ID, which Thoroughbred Basic uses to reference the device. Valid values are two characters long. In most cases, the first character is a **T**. Valid values for the second character are **0** through **9**, **A** through **Z**, and **a** through **z**.

type specifies the device type. Valid values are:

5 for a terminal that uses Thoroughbred Basic Windows.

6 for a terminal that uses a video card. Most systems running under MS-DOS require this specification.

7 for a terminal that does not use Thoroughbred Basic Windows.

To configure serial ports when you use the TTY*n* device under MS-DOS or Microsoft Windows, specify **7**.

baud is a positional parameter that specifies baud rate. Most terminal devices do not require you to specify a value for this parameter.

To configure serial ports when you use the TTY*n* device under MS-DOS or Microsoft Windows, you can specify one of the following values for *baud*:

110
150
300
600
1200
2400
4800
9600
14400
19200
38000
56000
115200
128000
256000

The default is the value specified by the MS-DOS MODE command.

parity is a positional parameter that specifies parity. Most terminal devices do not require you to specify a value for this parameter.

To configure serial ports when you use the TTY*n* device under MS-DOS or Microsoft Windows, you can specify one of the following values for *parity*:

0 No parity
1 Odd parity
2 Even parity

The default is the value specified by the MS-DOS MODE command.

width is a positional parameter that specifies width. Most terminal devices do not require you to specify a value for this parameter.

To configure serial ports when you use the TTY*n* device under MS-DOS or Microsoft Windows, You can specify one of the following values for *width*:

7
8

The default is the value specified by the MS-DOS MODE command.

stopbits is a positional parameter that specifies the number of stop bits. Most terminal devices do not require you to specify a value for this parameter.

To configure serial ports when you use the TTY*n* device under MS-DOS or Microsoft Windows, you can specify one of the following values for *stopbits*:

1
2

The default is the value specified by the MS-DOS MODE command.

nullstrip is a positional parameter that specifies how null characters are treated and whether bit 8 will be stripped. Most terminal devices do not require you to specify a value for this parameter.

To configure serial ports when you use the TTY*n* device, you can specify one of the following values for *nullstrip*:

0 means discard nulls and strip bit 8. This value is valid under Microsoft Windows or MS-DOS.

1 means discard nulls but do not strip bit 8. This value is valid under Microsoft Windows or MS-DOS.

2 means do not discard nulls but strip bit 8. This value is valid under UNIX.

3 means do not discard nulls and do not strip bit 8. This value is valid under UNIX.

Under Microsoft Windows or MS-DOS, the default is the value specified by the MS-DOS MODE command. The value can be **0** or **1**.

driver|port is a positional parameter that specifies the name of a terminal driver or system port. You must specify a value for this parameter.

Examples of the **DEV** *statement for terminal devices*

Example: **DEV** statement for a terminal

DEV T0,7,,,,,,tty

DEV	Begins the **DEV** statement.
DEV **T0**	Specifies that the device is called **T0**. You must make sure that this device name does not conflict with a task ID.
DEV T0,**7**	Specifies the device type for **T0**. Valid values are: **1** logical disk directory **4** standard dot matrix, character printer, or laser printer **5** terminal that uses Thoroughbred Basic Windows **6** terminal that uses a video card **7** terminal that does not use Thoroughbred Basic Windows **8** ghost task **T0** is defined as a terminal that will not use Thoroughbred Basic Windows.
DEV T0,7,,,,,,**tty**	Specifies the **tty** terminal driver. In most cases, **tty** is a generic terminal driver. Thoroughbred Basic will change the **T0** device name based on the position of the **tty** specification in the TERMINAL file. For more information on the TERMINAL file, please refer to Section 8.3. **Note:** If **5** were specified as a device type the terminal would be defined as a terminal that uses Thoroughbred Basic Windows. In this case, Thoroughbred Basic uses the Thoroughbred Basic Windows Manager to communicate with the terminal. Thoroughbred Basic Windows enables you to create a windowing interface for your application. It may slow terminal output. Thoroughbred Basic Windows may have display problems on terminals where character attributes occupy a physical position on the screen. For more information on Thoroughbred Basic Windows, please refer to the Thoroughbred Basic Developer Guide.

Example: **DEV** statement for a monitor under MS-DOS

DEV T0,6,,,,,,CON

DEV	Begins the **DEV** statement.
DEV **T0**	Specifies that the device is called **T0**. You must make sure that this device name does not conflict with a task ID.
DEV T0,**6**	Specifies the device type for **T0**. Valid values are: **1** logical disk directory **4** standard dot matrix, character printer, or laser printer **5** terminal that uses Thoroughbred Basic Windows **6** terminal that uses a video card **7** terminal that does not use Thoroughbred Basic Windows **8** ghost task **T0** is defined as a monitor that uses a video card. This monitor will not run Thoroughbred Basic Windows. **Note:** Under MS-DOS, device types **6** and **7** are roughly equivalent. Neither specification enables you to use Thoroughbred Basic Windows. To use Thoroughbred Basic Windows under MS-DOS you must specify **5** as the device type. For more information on Thoroughbred Basic Windows, please refer to the Thoroughbred Basic Developer Guide.
DEV T0,6,,,,,,**CON**	Specifies that the name of the terminal is **CON**, which is short for console. Under MS-DOS, you can specify communication ports in place of **CON**. You can specify **COM1** or **COM2** for standard communication. For interrupt-driven communication support, you can specify **TTY1** or **TTY2**.

Example: **DEV** statement to configure a serial port when using the TTY*n* device under MS-DOS or Microsoft Windows

DEV T0,7,9600,2,8,1,1,TTY1

DEV	Begins the **DEV** statement.
DEV **T0**	Specifies that the device is called **T0**. You must make sure that this device name does not conflict with a task ID.

DEV T0,**7**

Specifies the device type for **T0**. Valid values are:

1 logical disk directory

4 standard dot matrix, character printer, or laser printer

5 terminal that uses Thoroughbred Basic

6 terminal that uses a video card

7 terminal that does not use Thoroughbred Basic Windows

8 ghost task

T0 is defined as a terminal that does not run Thoroughbred Basic Windows. To configure a serial port when using the TTY*n* device under MS-DOS or Microsoft Windows, you must specify **7**.

DEV T0,7,**9600**

Specifies the baud rate. Valid values are:

110
150
300
600
1200
2400
4800
9600
14400
19200
38000
56000
115200
128000
256000

Baud is set to **9600**.

DEV T0,7,9600,**2**

Specifies parity. Valid values are:

0 No parity
1 Odd parity
2 Even parity

Parity is set to even parity.

DEV T0,7,9600,2,**8**	Specifies width, the number of data bits in a character. Valid values are: **7** **8** Width is set to **8** bits.
DEV T0,7,9600,2,8,**1**	Specifies the number of stop bits. Valid values are: **1** **2** The number of stop bits is set to **1**.
DEV T0,7,9600,2,8,1,**1**	Specifies how nulls are treated and whether bit 8 is stripped. Valid values are: **0** means discard nulls and strip bit 8. This value is valid under Microsoft Windows or MS-DOS. **1** means discard nulls but do not strip bit 8. This value is valid under Microsoft Windows or MS-DOS. **2** means do not discard nulls but strip bit 8. This value is valid under UNIX. **3** means do not discard nulls and do not strip bit 8. This value is valid under UNIX. Null characters will be discarded, but bit 8 will not be stripped.
DEV T0,7,9600,2,8,1,1,**TTY1**	Specifies the name of the TTY*n* device as **TTY1**.

Example: **DEV** statement for a terminal-oriented device

DEV T1,7,,,,,,**tty05b**

DEV	Begins the **DEV** statement.
DEV **T1**	Specifies that the device is called **T1**. You must make sure that this device name does not conflict with a task ID.

DEV T1,**7** Specifies the device type for **T1**. Valid values are:

1 logical disk directory
4 standard dot matrix, character printer, or laser printer
5 terminal that uses Thoroughbred Basic Windows
6 terminal that uses a video card
7 terminal that does not use Thoroughbred Basic Windows
8 ghost task

T1 is defined as a monitor that will not run Thoroughbred Basic Windows.

DEV T1,,,,,,**tty05b** Specifies the tty05b system port.

This specification enables this task to communicate to another terminal or to a modem attached to that port. This type of terminal-oriented device must be specified with a device type of **7**. It cannot use the Thoroughbred Basic Windows Manager.

8.5.8 DEV statement syntax for a printer

DEV *dev-ID*,*type*,*param-1*,*width*,*lockflag*, *spooler*,*time-out*,*devicename*

DEV begins the **DEV** statement. The statement begins in the leftmost column.

dev-ID is the device ID, which Thoroughbred Basic uses to reference the printer device. In most cases, you will use one of the following formats to specify *dev-ID*:

L*X* is a printer device. Printer devices include dot matrix printers, character printers, laser printers, print spoolers, and slave printers. Valid values for *X* are **0** through **9**, **A** through **Z**, and **a** through **z**.

P*X* is a printer device. Printer devices include dot matrix printers, character printers, laser printers, print spoolers, and slave printers. Valid values for *X* are **0** through **9**, **A** through **Z**, and **a** through **z**.

type specifies the device type. For a printer device, the only valid value is **4**.

param-1 is a positional parameter. For printer devices, do not specify a value for this parameter.

width is a positional parameter that specifies the line width. The largest valid value is the maximum number of characters that can be printed on one line. The default is **132**.

lockflag is a positional parameter that specifies whether the printer will be available to more than one task at the same time. Valid values are:

0 Printer locking is in effect. The printer is only available to one task at a time. This is the default.

Thoroughbred Basic, not the operating system, maintains printer locking. Under UNIX, when an **OPEN** directive is issued, Thoroughbred Basic creates two files, one to link the printer to the task and one to lock the printer. In the case of a printer named **lp** and a task ID named **T5**, the files would be named **/tmp/lp.T5** and **/tmp/lp.lck**. When the **CLOSE** directive is issued, Thoroughbred Basic erases these two files.

1 Printer sharing is in effect. The printer is available to more than one task at a time.

Because a value of **1** enables multiple tasks to use the printer at the same time, this specification can result in merged output printing. In most cases, this specification is used for spooled printing.

For more information on the directives mentioned above, please refer to the Thoroughbred Basic Language Reference.

spooler is a positional parameter that specifies what type of printer device is associated with the *dev-ID*. Valid values are:

0 A directly connected printer. There is no spooling. This is the default.

The name of the printer is specified in the *devicename* parameter.

1 A spooler that uses piped input. A pipe is created between Thoroughbred Basic and an executable program specified by the *devicename* parameter. Output will be sent through the pipe to the program. The program can be a shell script or a binary executable.

Under Microsoft Windows

Under Microsoft Windows, this specifies pass-through Print Manager mode. All output will be spooled to the Microsoft Windows Print Manager. Pass-through mode uses existing Dictionary-IV printer tables. If you use the Windows Print Setup facility to change printer type, you must reassign your printer table to the new type.

Programs that use escape sequences or mnemonics that the Print Manager does not recognize will generate errors.

To use the Print Setup facility, click the File option on the menu bar. For more information on Dictionary-IV printer tables, please refer to the Dictionary-IV Reference Manual.

2 A slave printer, which is a printer connected to a terminal's auxiliary or printer port and available only to that terminal. Output can be sent to the secondary port by using the **'PS'** and **'PE'** terminal mnemonics. For more information on slave printers and terminal mnemonics, please refer to Section 3.4.

The terminal device is specified in the *devicename* parameter.

3 A spooler that uses a temporary file. Thoroughbred Basic will create a temporary file to contain output. After the file is closed, its contents will be released to an executable program specified by the *devicename* parameter. The program can be a shell script or a binary executable.

Under Microsoft Windows

Under Microsoft Windows, **3** specifies standard Print Manager mode. All output will be spooled to the Microsoft Windows Print Manager. Standard mode uses a standard printer table, which enables you to use the Windows Print Setup facility to change printer type without having to reassign your printer table to the new type.

To use a standard printer table, you must create a printer table that contains a set of escape sequences. For more information on how to create the printer table, please refer to the Thoroughbred Environment for Windows Supplemental Guide.

Programs that use escape sequences or mnemonics that the Print Manager does not recognize will generate errors.

To use the Print Setup facility, click the File option on the menu bar. For more information on Dictionary-IV printer tables, please refer to the Dictionary-IV Reference Manual.

4 Under Microsoft Windows, specifies an append to a file.

time-out is a positional parameter that specifies the printer time-out. Valid values are numbers, which specify seconds. If the printer does not respond within the specified number of seconds, Thoroughbred Basic generates a time-out error. The default is **15** seconds.

devicename is a positional parameter that contains the name of a printer device, terminal device, system port, or program. For more information on valid specifications, please refer to the description of the *spooler* parameter.

Examples of the **DEV** *statement for printer devices*

Example: **DEV** statement for a printer

DEV LP,4,,142,1,1,,lp -dP4

DEV	Begins the **DEV** statement.
DEV **LP**	Specifies that the device is called **LP**.
DEV LP,**4**	Specifies the device type for **LP**. Valid values are:

1 logical disk directory

4 standard dot matrix, character printer, or laser printer

5 terminal that uses Thoroughbred Basic Windows

6 terminal that uses a video card

7 terminal that does not use Thoroughbred Basic Windows

8 ghost task

LP is defined as a printer.

DEV LP,4,,**142** Specifies the maximum line width. In this example, the maximum line width is **142** characters. The default is **132**.

DEV LP,4,,142,**1** Specifies the printer lock flag. Valid values are:

0 The printer is only available to one task at a time. This is the default.

1 The printer is available to more than one task at a time.

Because a value of **1** enables multiple tasks to use the printer at the same time, this specification can result in merged output printing. In most cases, this specification is used for special printers or point-of-sale stations.

Thoroughbred Basic, not the operating system, maintains printer locking. When an **OPEN** directive is issued, Thoroughbred Basic creates two files, one to link the printer to the task and one to lock the printer. In the case of a printer named **lp** and a task ID named **T5**, the files would be named **/tmp/lp.T5** and **/tmp/lp.lck**. When the **CLOSE** directive is issued, Thoroughbred Basic erases these two files.

For more information on the directives mentioned above, please refer to the Thoroughbred Basic Language Reference.

DEV LP,4,,142,1,**1** Specifies the print spooler flag. Valid values are:

0 A directly connected printer. There is no spooling. This is the default.

1 A spooler that uses piped output

2 A slave printer, which is attached to printer port of the terminal this task is using.

3 A spooler. Thoroughbred Basic will create a **/tmp** file. After the file is closed, it will be released to the spooler.

4 Under Microsoft Windows, specifies an append to a file.

LP will use a spooler that expects piped output.

DEV LP,4,,142,1,1,,**lp -dP4** Specifies **lp** as the system name of the print spooler and **-dP4** as the parameters that will be sent to the spooler with the piped output.

For more information on print spoolers and print parameters, please refer to your operating system documentation.

Example: **DEV** statement for a slave printer

DEV P1,4,,132,,2,5,tty

DEV Begins the **DEV** statement.

DEV **P1** Specifies that the device is called **P1**.

DEV P1,**4** Specifies the device type for **P1**. Valid values are:

1 logical disk directory
4 standard dot matrix, character printer, or laser printer
5 terminal that uses Thoroughbred Basic Windows
6 terminal that uses a video card
7 terminal that does not use Thoroughbred Basic Windows
8 ghost task

P1 is defined as a printer.

DEV P1,4,,**132** Specifies the maximum line width. In this example, the maximum line width is **132** characters, which is also the default.

DEV P1,4,,132,,**2** Specifies the print spooler flag. Valid values are:

0 A directly connected printer. There is no spooling. This is the default.

1 A spooler that uses piped output

2 A slave printer, which is attached to printer port of the terminal this task is using.

3 A spooler. Thoroughbred Basic will create a **/tmp** file. After the file is closed, it will be released to the spooler.

4 Under Microsoft Windows, specifies an append to a file.

P1 is defined as a slave printer. For more information on slave printers, please refer to Chapter 3.

DEV P1,4,,132,,2,**5** Specifies the printer response time. In this example, the time is **5** seconds. The default is **15**.

The printer will generate a time-out error if it does not respond within 5 seconds.

DEV P1,4,,132,,2,5,**tty** Specifies the generic terminal driver. A slave printer can only be used by the task associated with the terminal. It is not classed as a system printer.

Example: **DEV** statement for pass-through Print Manager mode under Microsoft Windows

DEV LP,4,,,,1,,LPT1

DEV Begins the **DEV** statement.

DEV **LP** Specifies that the device is called **LP**.

DEV LP,**4** Specifies the device type for **LP**. Valid values are:

1 logical disk directory
4 standard dot matrix, character printer, or laser printer
5 terminal that uses Thoroughbred Basic Windows
6 terminal that uses a video card
7 terminal that does not use Thoroughbred Basic Windows
8 ghost task

LP is defined as a printer.

DEV LP,4,,,,**1** Specifies the print spooler flag. Valid values are:

0 A directly connected printer. There is no spooling. This is the default.

1 A spooler that uses piped output

2 A slave printer, which is attached to printer port of the terminal this task is using.

3 A spooler. Thoroughbred Basic will create a **/tmp** file. After the file is closed, it will be released to the spooler.

4 Under Microsoft Windows, specifies an append to a file.

LP is defined as a printer that expects spooled output.

Under Microsoft Windows, **1** specifies pass-through Print Manager mode. All output would be spooled to the Microsoft Windows Print Manager. Pass-through mode uses existing Dictionary-IV printer tables. If you use the Windows Print Setup facility to change printer type, you must reassign your printer table to the new type.

Programs that use escape sequences or mnemonics that the Print Manager does not recognize will generate errors.

To use the Print Setup facility, click the File option on the menu bar. For more information on Dictionary-IV printer tables, please refer to the Dictionary-IV Reference Manual.

DEV LP,4,,,,1,,**LPT1** Specifies the **LPT1:** system port. This example assumes that the Microsoft Windows Print Manager will spool output to this parallel port.

Example: **DEV** statement for standard Print Manager mode under Microsoft Windows

DEV LP,4,,,,3,,LPT1

DEV Begins the **DEV** statement.

DEV **LP** Specifies that the device is called **LP**.

DEV LP,**4** Specifies the device type for **LP**. Valid values are:

1 logical disk directory
4 standard dot matrix, character printer, or laser printer
5 terminal that uses Thoroughbred Basic Windows
6 terminal that uses a video card
7 terminal that does not use Thoroughbred Basic Windows
8 ghost task

LP is defined as a printer.

DEV LP,4,,,,**3** Specifies the print spooler flag. Valid values are:

0 A directly connected printer. There is no spooling. This is the default.

1 A spooler that uses piped output

2 A slave printer, which is attached to printer port of the terminal this task is using.

3 A spooler. Thoroughbred Basic will create a **/tmp** file. After the file is closed, it will be released to the spooler.

4 Under Microsoft Windows, specifies an append to a file.

LP is defined as a spooler.

Under Microsoft Windows, **3** specifies standard Print Manager mode. All output will be spooled to the Microsoft Windows Print Manager. Standard mode uses a standard printer table, which enables you to use the Windows Print Setup facility to change printer type without having to reassign your printer table to the new type.

To use a standard printer table, you must create a printer table that contains a set of escape sequences. For more information on how to create the printer table, please refer to the Thoroughbred Environment for Windows Supplemental Guide.

Programs that use escape sequences or mnemonics that the Print Manager does not recognize will generate errors.

To use the Print Setup facility, click the File option on the menu bar. For more information on Dictionary-IV printer tables, please refer to the Dictionary-IV Reference Manual.

DEV LP,4,,,,3,,**LPT1** Specifies the **LPT1:** system port. This example assumes that the Microsoft Windows Print Manager will spool output to this parallel port.

Example: **DEV** statement for printer and intelligent I/O communications board

DEV P2,4,,,,,,ttp8e

DEV Begins the **DEV** statement.

DEV **P2** Specifies that the device is called **P2**.

DEV P1,**4** Specifies the device type for **P2**. Valid values are:

1 logical disk directory
4 standard dot matrix, character printer, or laser printer
5 terminal that uses Thoroughbred Basic Windows
6 terminal that uses a video card
7 terminal that does not use Thoroughbred Basic Windows
8 ghost task

P2 is defined as a printer.

DEV P2,4,,,,,,**ttp8e** Specifies a system port named **ttp8e**. By default, the line width is 132 characters, the printer is directly connected to the system port, and the printer response time is 15 seconds.

Note: Some manufacturers of intelligent I/O communication boards provide the ability to address a printer attached to the back of a terminal by using separate drivers for the I/O board.

In this example, the **ttp8e** port name is assigned by the Stargate Serial I/O Xenix Drivers for the Stargate ACL board. The board specifies a slave printer attached to a terminal connected to the tty8e serial port, which enables the printer to be a system printer. The printer is available to any task provided this **DEV** statement is specified in its associated IPL file.

8.5.9 DEV statement syntax for a ghost task

DEV G*X*,*type*

DEV begins the **DEV** statement. The statement begins in the leftmost column.

G*X* is the device ID, which Thoroughbred Basic uses to reference the ghost task. Valid values are **G0** through **G9** and **GA** through **GZ**. You can specify up to 36 ghost tasks.

Ghost task device specifications must follow ascending order. For example, if you plan to specify 36 ghost tasks you must specify **G0** through **G9**, then **GA** through **GZ**.

type specifies the device type. For a ghost task, the only valid value is **8**.

Examples of the **DEV** *statement for ghost tasks*

Example: **DEV** statement for the first ghost task

DEV G0,8

DEV	Begins the **DEV** statement.
DEV **G0**	Specifies that the device is called **G0**. The only valid names for ghost tasks are **G0** through **G9** and **GA** through **GZ**. The first ghost task defined in an IPL file must be named **G0**.
DEV G0,**8**	Specifies the device type for **G0**. Valid values are: **1** logical disk directory **4** standard dot matrix, character printer, or laser printer **5** terminal that uses Thoroughbred Basic Windows **6** terminal that uses a video card **7** terminal that does not use Thoroughbred Basic Windows **8** ghost task **G0** is defined as a ghost task. For more information on ghost tasks, please refer to Chapter 5 of this manual and to the Thoroughbred Basic Developer Guide.

Example: **DEV** statement for the second ghost task

DEV G1,8

DEV	Begins the **DEV** statement.
DEV **G1**	Specifies that the device is called **G1**. The only valid names for ghost tasks are **G0** through **G9** and **GA** through **GZ**.

The first ghost task defined in an IPL file must be named **G0** and the second must be named **G1**.

DEV G1,**8** Specifies the device type for **G1**. Valid values are:

1 logical disk directory
4 standard dot matrix, character printer, or laser printer
5 terminal that uses Thoroughbred Basic Windows
6 terminal that uses a video card
7 terminal that does not use Thoroughbred Basic Windows
8 ghost task

G1 is defined as a ghost task.

Note: If you plan to use more than one IPL file for one Thoroughbred Basic system, each file must contain identical ghost task specifications.

For example, if the IPLINPUT file contains **DEV** statements for five ghost tasks named **G0** through **G4**, every other IPL file must contain those **DEV** statements. No IPL file can define more than five ghost tasks.

8.6 The IPL statement

The **IPL** statement specifies the initial program this task will load when Thoroughbred Basic starts. An IPL file contains only one **IPL** statement, which must occupy the next to last line in the file. It must be placed between the **DEV** statements and the **END** statements.

8.6.1 Overview of the IPL statement

The **IPL** statement has the following format:

IPL {*partnum*}**,***reserved***,**{*terminal*}**,***progname*

IPL begins the IPL statement.

partnum is the partition number. The only valid value is **1**.

reserved is a parameter reserved for Thoroughbred Basic internal use. Thoroughbred Basic will ignore a numeric specification.

terminal is the **DEV** statement name in this IPLINPUT file that defines the terminal to associate with this task. This name must match the **DEV** statement name. Thoroughbred Basic will change this name, and the name specified in the **DEV** statement, to the appropriate task ID.

progname is the program to run when this task is initiated. The default is to leave the task in Thoroughbred Basic Console Mode with no program loaded.

8.6.2 Example of the IPL statement

IPL 1,2,T0,PSD**

The partition number is **1**. The value of the reserved parameter is **2**, but Thoroughbred Basic will ignore this specification. The terminal associated with this task is **T0**. When Thoroughbred Basic starts, it will load and run the ****PSD** program.

8.7 The END statement

The **END** statement specifies the end of the IPL file. An IPL file contains only one **END** statement, which must occupy the last line in the file.

The **IPL** statement has the following format:

END

The **END** statement has no parameters.

8.8 Environment variables

Thoroughbred Basic under UNIX or Thoroughbred Environment under Microsoft Windows enables you to specify environment variables in the IPL file. Environment variables are appropriately changed or expanded when Thoroughbred Basic or Thoroughbred Environment starts.

8.8.1 Overview of environment variables

Specify an environment variable by placing a **$** in front of the name of the variable defined in the operating system. Environment variables names are case sensitive. For example, the Microsoft Windows **windir** environment variable is usually specified as **windir=C:\WINDOWS**. To specify the environment variable in the IPL file, type **$windir**.

An invalid environment variable name specification in the IPL file will be removed. For example, if **$WIN2DOOR** is specified in the IPL file but **WIN2DOOR** is not an environment variable, the **$WIN2DOOR** specification will be removed when Thoroughbred Basic or Thoroughbred Environment starts.

To display active environment variables under UNIX or Microsoft Windows, use the **set** command. Under Microsoft Windows, you can issue this command in the MS-DOS Prompt window.

8.8.2 Examples of environment variable specification

Example of valid environment variable specification:

Given the following environment variable:

STARTPRG=INITWORK

And the following statement in an IPL file:

IPL 1,2,T0,$STARTPRG

The statement is changed to the following when Thoroughbred Basic or Thoroughbred Environment starts:

IPL 1,2,T0,INITWORK

Example of invalid variable specification:

Given the following statement in an IPL file:

IPL 1,2,T0,$STARTPRG

If the **STARTPRG** environment variable name is invalid, the statement is changed to the following when Thoroughbred Basic or Thoroughbred Environment starts:

IPL 1,2,T0,

Example for Thoroughbred Environment:

Given the following environment variable:

windir=C:\WINDOWS

And the following statement in an IPL file:

DEV D0,1,,,,,,$windir

The statement is changed to the following when Thoroughbred Basic or Thoroughbred Environment starts:

DEV D0,1,,,,,,C:\WINDOWS

Example for UNIX:

A UNIX **$** variable is a portion of a line to the left of the **=** sign, from the UNIX **env** command, followed by a **$**.

For example, given the following **env**:

=bin/env
HOME=/user/scott
PWD=/usr/lib/basic
SHELL=/bin/ksh
MAIL=/usr/mail/scott
EDITOR=vi
LOGNAME=scott
LPDEST=local
PS2=<continued>
PS1=! hp840 $PWD $
TERM=wy60
PATH=:/usr/lbin:/bin:/usr/bin:/user/scott:.
TZ=EST5EDT

And the following IPL file:

```
CNF 1,8,1,45
PTN 1,200000
PRM LONGVAR
PRMIF47
PRM UPPER
PRM SYSTEM=$SHELL
DEV D0,1,,,,,,$PWD/UTILS
IPL 1,2,T0
END
```

The IPL file expands to the following when Thoroughbred Basic starts:

```
CNF 1,8,1,45
PTN 1,200000
PRM LONGVAR
PRM IF47
PRM UPPER
PRM SYSTEM=/bin/ksh
DEV D0,1,,,,,,/usr/lib/basic/UTILS
IPL 1,2,T0
END
```

9. The IPLPRM File

This operating system file defines Thoroughbred Basic environment parameters that apply to all users of Thoroughbred Basic. This file may contain any of the **PRM** statements described in section 8.4.2 of this manual, optionally followed by an **END** statement. The **PRM** statements in this file are inserted immediately after the **PTN** statement while the **IPLINPUT** file is being loaded into memory. When the same PRM statement appears in both the global **IPLPRM** file and the local **IPLINPUT** file, the action taken is the same as if both statements were present in **IPLINPUT**.

Note: The IPLPRM can be disabled by setting the IPLPRM variable to **N** on individual systems. Example:
IPLPRM=N

9.1 Sample IPLPRM File

```
PRM OPENLIB=IDOL-IV,32700
PRM PGCHARBASE=8
PRM JOURNALING
END
```

9.2 Location of IPLPRM Files

The name and location of the IPLPRM file varies by Operating System:

Operating System	Location
UNIX Systems	/usr/lib/basic/IPLPRM
Microsoft Windows	C:\WINDOWS\IPLPRM.TXT or C:\WINNT\IPLPRM.TXT
VMS	TBRED_BASIC:IPLPRM